IDA

INSTITUTE FOR DEFENSE ANALYSES

FY96 Analysis of the Ballistic Missile Defense Interoperability Standards

Dennis W. Fife, Task Leader
Reginald N. Meeson
David A. Wheeler

September 1996

IDA Paper P-3277
Log : H 96-003846

This work was conducted under contract DASW01 94 C 0054, Task T-R2-597.2, for the Ballistic Missile Defense Organization. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

PREFACE

This annotated briefing was prepared by the Institute for Defense Analyses (IDA) under IDA task order T-R2-597.2 and was sponsored by the Ballistic Missile Defense Organization (BMDO). It fulfills one objective of the FY 1996 task order, "to prepare an informal report comprising a Ballistic Missile Defense (BMD) battle management, command, control, and communications (BMC3) interoperability standards plan."

The following research staff members were reviewers of this document: Dr. Richard Ivanetich, Dr. Alfred Brenner, Dr. Norman Howes, Dr. Edward Feustel, and Dr. Richard Morton.

EXECUTIVE SUMMARY

The purpose of this annotated briefing is to (1) present an assessment of needs for interoperability standards and (2) recommend actions that BMDO should take based on that assessment. The purpose of BMC3 interoperability standards is to specify capabilities that BMD systems must have to interact and adapt as needed to meet mission goals.

This presentation highlights issues in using the DoD Joint Technical Architecture (JTA) for BMD BMC3. These issues were identified by examining selected joint BMD scenarios, the standards needed to support these scenarios, and the state of relevant standards and suites of standards. The presentation identifies additional open standards needed in BMD BMC3 systems and identifies incompatibilities and limitations of the JTA. It is not intended to select or recommend a set of BMD BMC3 interoperability standards, since the JTA has started to fill that need (though more remains to be done). The scope of this presentation is limited to information technology standards; other standards (such as form and fit) were not considered.

The recommendations are divided into three areas: (1) standards selection and tracking, (2) interoperability architecture development, and (3) compliance and migration planning. In the first area, we recommend significant effort to monitor the maturity and usability of the relevant JTA standards and emerging technologies. We also believe key standards and products need accelerating action, e.g., security application program interfaces (APIs), Common Operating Environment (COE) APIs, internet protocol (IP) quality of service, and combat net radio (CNR) IP mobility. In the second area, we recommend the improvement of a BMD technical architecture beyond simply a list of standards, to identify where and how standards will be applied. Specific work is needed to define BMD domain-specific open system protocols and to separate messages from their communications medium. In the third area, we recommend that BMDO and the Services coordinate their compliance efforts to avoid duplication of standards tracking and prototyping efforts. Compliance progress assessment should emphasize the positive benefits from lessons learned and the identification of methods for handling unexpected difficulties, not on "generating scores." Compliance is an evolutionary process and should be based on long-term criteria.

INSTITUTE FOR DEFENSE ANALYSES
Paper P-3277

FY96 ANALYSIS OF THE BALLISTIC MISSILE DEFENSE INTEROPERABILITY STANDARDS

Dennis Fife
Reg Meeson
David A. Wheeler

30 September 1996

IDA

OUTLINE OF INTEROPERABILITY STANDARDS PLAN

This chart and the next outline the content of the interoperability standards plan as presented in this report. The Purpose and Scope describe the plan's purpose and scope, and the objectives of BMC3 interoperability standards. This plan is not intended to select or recommend a set of BMD BMC3 interoperability standards. The DoD Joint Technical Architecture (JTA) Version 1 [DoD 1996] has started to fill that need, though more remains to be done. This plan instead highlights issues in using the JTA for BMD BMC3 and provides recommendations on future actions that may resolve such issues.

Background reviews DoD events that transpired in the area of Technical Architecture during this work and influenced its outcome. Also, IDA's perspective on interoperability and closely related qualities such as mission adaptability is explained in the Background.

Analysis of joint ballistic missile defense scenarios is a primary tool for identifying and understanding BMC3 interoperability issues. Here, we conceived a series of scenarios for multi-tiered Theater Missile Defense (TMD) and for National Missile Defense (NMD) to examine specific (though hypothetical) interoperability opportunities. The conclusions reveal where missile defense domain standards could be beneficial to extending BMD capability beyond the stand-alone capabilities of individual weapon and sensor systems.

The next chart continues the plan outline.

OUTLINE OF INTEROPERABILITY STANDARDS PLAN

Purpose and Scope (page 6)

- Purpose and scope of plan
- Objectives of BMC3 interoperability standards

Background on Technical Architecture and Interoperability (page 8)

Joint BMD Evolution Scenarios (page 10)

- Purpose, goals, and basis for formulating joint scenarios
- Description and analysis of scenarios
- Scenario findings on interoperability needs

OUTLINE (CONTINUED)

The next section of the plan assesses interoperability standards, emphasizing general mechanisms for interoperability and distributed processing, such as the Distributed Computing Environment (DCE). Primary attention is focused on mechanisms accepted in the Joint Technical Architecture. Typical issues of completeness, options, and compatibility of recommended standards, as well as maturity, compliance, and replaceability of proprietary products are addressed. The analysis answers how complete and detailed the guidance accompanying a set of standards must be to meet its purpose.

The last section of the plan recommends important principles for BMDO's process to evaluate and select needed standards, and to achieve the results desired from standards. Standards selection considers the need for actions and expertise to deal with technology and marketplace evolution as well as JTA evolution. Interoperability architecture development concerns principles and actions for exploiting DoD interoperability guidance and achieving a joint BMD interoperability architecture. Compliance and migration planning deals with the need, interpretation, and available means to achieve compliance of BMD systems with a set of interoperability standards.

Appendix A provides extensive technical details on the concepts for three commercial standards that are gaining importance for implementing communications among distributed processing applications on a network. The standards are the Open Group's Distributed Computing Environment (DCE), the Object Management Group's (OMG's) Object Management Architecture (OMA) including its Common Object Request Broker Architecture (CORBA), and Microsoft's Component Object Model (COM).

OUTLINE (CONTINUED)

PURPOSE AND SCOPE

This chart states the purposes for this standards plan, and the reason that BMD needs interoperability standards. The scope of this effort is stated in terms of specific objectives that were chosen at the outset.

PURPOSE AND SCOPE

- Purposes of standards plan
  - Assess needs that may exist for interoperability standards across BMD systems
  - Recommend actions that BMDO may take to select and apply effective BMC3 standards
- Purpose of BMC3 interoperability standards
  - Specify capabilities that BMD systems must have (in time) so as to interact and adapt as needed to meet mission goals
- Scope of effort
  - Demonstrate value of addressing concrete mission needs
  - Identify JTA standards of concrete value to BMD evolution
  - Identify additional open standards needed in BMD BMC3
  - Identify incompatibilities, limitations of JTA & open standards
  - Identify migration problems in BMD compliance to JTA

BACKGROUND

This chart briefly states several considerations that have influenced this result, such as the evolution of the Joint Technical Architecture, an effort that began about mid-way through FY 1996. The efforts that began as the C4ISR Integration Task Force, although not having a major impact on this result, may have more influence in the future on BMD architectures, and so must be monitored. At least, the FY 1996 report of its Integrated Architectures Panel has stressed the value of three architecture views to DoD decision making. This plan considers interoperability in the light of these DoD-recognized architecture views, i.e., Technical Architecture (TA), System Architecture (SA), and Operational Architecture (OA). An important principle is that interoperability requires more than communication connectivity. It depends also on comparable command and control operations and chain of command structure, as would be expressed in an OA that describes required interoperation.

In addressing interoperability needs, the IDA project staff have been keenly aware of the limitations of interconnection standards alone as a solution to BMD needs. In addition to the capability to transport data in "messages" of various types, interoperability requires that systems have mutual understanding of the behavior that is expected in response to defined message types. This means that messages must be specified in the context of a protocol defining the behavior involved. For real-time applications such as BMD or other weapon systems, timing requirements are an important part of protocols.

The JTA does not designate protocols but they are defined or implicit in many of its mandated standards, e.g., those for data communications and networking. Mission oriented protocols, as needed by BMD BMC3 and other DoD C4I applications, are implicit in application and mission support software. Thus the Defense Information Infrastructure Common Operating Environment (DII COE) implicitly involves a number of protocols. Those will have to be made explicit in future COE standardization activities so that interoperability can be achieved across different COE implementations.

The IDA project staff have also been concerned about other goals for standards that are important to warfighting capability. Primary examples of other goals include adaptability and configurability of mission applications and support services. Adaptability and configurability would permit the tailoring of standardized systems to the particular needs of a theater, mission, or commander. The scope of this effort was not addressed in this plan.

BACKGROUND

- When this effort began early in FY 1996,
  - Army was just updating its Technical Architecture to Version 3
  - DISA planned to release a TAFIM update (Version 3)
  - C4ISR Integration Task Force was not yet formed, nor the Joint Technical Architecture (JTA) Working Group
- As JTA results began to emerge, IDA staff participated in BMDO's internal tiger team review. JTA Version 1 was released (July), and mandated by OSD in August, 1996
- IDA perspective
  - Interoperability requires more than capability to transport and receive standardized messages
  - Interoperability standards must include specification of "protocols" associated with standard messages, i.e., expected timing and behavior in response
  - Standards also must provide for adaptability and configurability, to best meet mission and commander demands

JOINT BALLISTIC MISSILE DEFENSE EVOLUTION SCENARIOS

This study was undertaken to answer the following questions posed by BMDO's Deputy for BMC3: What is missing from the JTA? What needs to be added for BMC3 (e.g., for real-time engagement)?

To answer these questions we generated several missile defense scenarios and attempted to extrapolate out beyond currently-planned system capabilities to explore possibilities for future enhanced system interoperability and adaptability. These extrapolations are hypothetical. In each case, the enhanced capabilities we considered involved interactions among component systems that are not part of their current specifications. This study considered only the technical issues of interface standards for these hypothetical scenarios. Other obvious issues such as treaty compliance were not considered.

No recommendations or decisions have been made to pursue any of these enhancements or even to estimate their feasibility or cost.

Our approach was to attempt to identify plausible mission capability enhancements and then identify standards that would help support them. Evolving system components toward these standards, therefore, would contribute toward potential development and fielding of advanced system capabilities.

JOINT BALLISTIC MISSILE DEFENSE EVOLUTION SCENARIOS

What is missing from the JTA?

What needs to be added for BMC3?

SCENARIOS EXAMINED

In this section we present five graduated scenarios showing increasing degrees of capability and interaction among BMD systems. Scenarios one through three examine Theater Missile Defense (TMD) scenarios, and scenarios four and five examine National Missile Defense (NMD) scenarios.

TMD has the potential to grow from initial fieldable systems that should be operational sometime in 1998 (Scenario 1), to an intermediate stage where all systems and system upgrades currently in development are added (Scenario 2, circa 2003), to an enhanced long-term capability where these systems collaborate in more flexible and more robust "mix-and-match" configurations (Scenario 3).

The fourth scenario depicts the initial National Missile Defense (NMD) system that should be operational sometime in 2003. Scenario 5 describes how TMD systems could be used to augment the NMD system to provide enhanced defensive capabilities.

As noted earlier, this study considered only the technical issues of interface standards for these hypothetical scenarios. Other obvious issues such as treaty compliance were not considered.

In each of the scenarios we identify and characterize the BMD systems, the communications infrastructure and standards they employ, and adaptability of those systems to changing mission requirements. In scenarios 3 and 5 we describe some of the mission enhancements that should be achievable if system interoperability and adaptability can be evolved toward identified standards that will take them well beyond their current specifications.

SCENARIOS EXAMINED

- Five Scenarios: 3 TMD, 2 NMD
- 1. Near-term TMD systems that will be operational ca. 1998
  - AEGIS, HAWK, PATRIOT (PAC 3)
- 2. Intermediate-term TMD systems operational ca. 2003
  - Add THAAD, upgrade AEGIS to NAD
- 3. Long-term TMD systems projected beyond 2003
  - Extend mission capabilities using existing components
- 4. Near-term NMD systems operational ca. 2003
  - Single- or multiple-site GBI, GBR, and BMC3
- 5. NMD augmented with TMD systems projected beyond 2003
  - Extend mission capabilities using existing components

JOINT TMD EVOLUTION SCENARIOS

We started our analysis with scenarios drawn from Theater Missile Defense. The types of interoperability and adaptability enhancements we were looking for among TMD systems and system components, would provide field commanders significantly more flexibility in configuring available resources and coordinating defensive operations. For example, we were looking for ways to combine resources of partially operating TMD systems to form fully operational defenses.

JOINT TMD EVOLUTION SCENARIOS

- Purpose and goals
  - Explore mission-driven interactions among TMD systems to identify interoperability and adaptability potential
  - Identify practical approaches for achieving increased interoperability and adaptability — "mix and match"
  - Identify where existing standards can be applied, areas where new (BMD-specific) standards are needed
- Basis for formulating scenarios
  - TMD C2 Plan and COEA system descriptions, system schedules
  - THAAD-PATRIOT engagement coordination activities
  - Future battlefield information infrastructure — ABIS, Global Grid

SCENARIO #1 (OPERATIONAL CA. 1998)

The first scenario involves only components that are currently fieldable or will be fieldable in the near future. For the most part, these systems were conceived and developed as independent, stand-alone systems. Each has its own "organic" sensors, interceptors, and battle management and command and control systems. These components are not compatible across systems. Each system has its own unique internal communications protocols and message sets.

Command and control data is transmitted to and from these systems, but there is no real-time battle data exchanged directly between systems. For example, a track observed by one PATRIOT battery will be reported to its Battalion Headquarters. If the threat's predicted impact point is outside that battery's defended area, the battery will not attempt to engage it. If the impact point lies within the area defended by another PATRIOT battery, it is up to the Battalion Headquarters to notify the second battery of the observed track. There is no direct communication between the two batteries. PATRIOT batteries do not monitor the reporting of neighboring batteries.

SCENARIO #1 (OPERATIONAL CA. 1998)

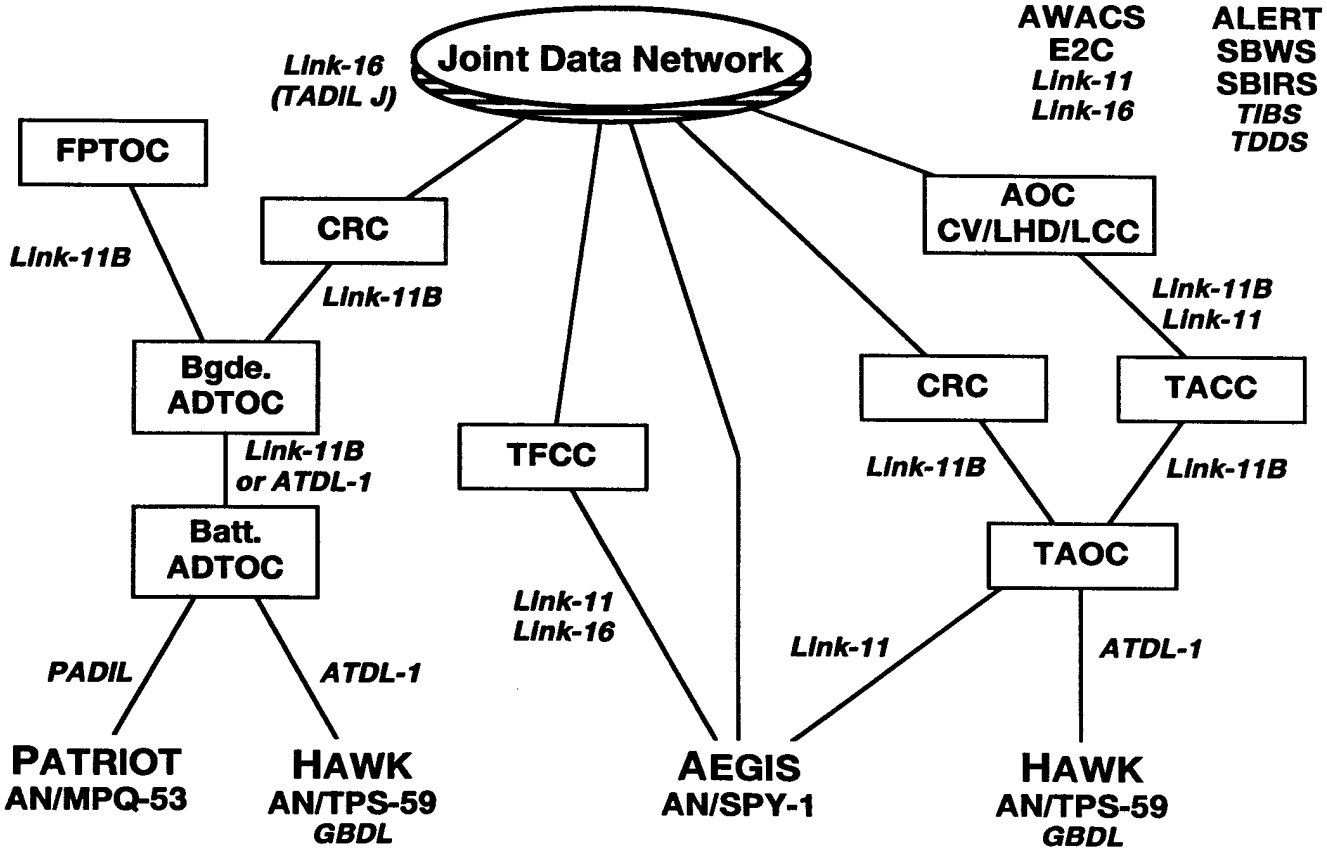
- AEGIS, HAWK, PATRIOT (PAC 3)
  - ALERT and SBWS warnings, AWACS and E2C track data
- Stand-alone systems, vertical communications
  - Sensors, interceptors, and BMC3 coupled by unique comms — forming closed systems (stovepipes)
  - Deployment locations, coverage areas, rules of engagement, intelligence, early warnings passed down
  - Status reported up, tactical picture data relayed to JDN
  - Locally autonomous engagement decisions within constraints
  - No real-time battle data exchanged among systems
- Adaptability
  - Limited changes in ROE and assignments during battle
  - Limited mix-and-match reconfiguration capabilities

COMMUNICATIONS FOR SCENARIO #1

This diagram shows the diversity of communications channels and protocols that would make up a Joint Theater Missile Defense system. Almost every organizational layer in each Service is connected by a different protocol and different message set. Numerous message conversions are required as information is passed up and down the chain of command.

The diagram also attempts to illustrate the vertical nature of information flow, at least at the lower echelons. Two PATRIOT batteries commanded by separate Battalions would have to exchange early warning track information at the Brigade level. The path that AEGIS track information might have to travel to get to a relevant PATRIOT battery is not clear. None of the paths appears to be direct or automatic.

COMMUNICATIONS FOR SCENARIO #1



IDA

STANDARDS FOR SCENARIO #1

Parts of systems in this first scenario are in compliance with the proposed BMD profile of modern standards. The largest category of non-compliant components is the legacy communications technology. Although this equipment continues to function, it provides limited bandwidth and restricts the legacy system's ability to communicate with other systems. If these systems are going to continue to play a role in theater defense, their communications subsystems will need to evolve.

According to the Joint Tactical Data Link Management Plan, several of these standards are due to be retired within this scenario's time frame (ca. 1998 — ATDL, GBDL, and PADIL).

STANDARDS FOR SCENARIO #1

- Standards in compliance with BMD standards profile
  - Ada, POSIX, Link-16 (TADIL J), MILSATCOM (TDDS, TIBS), C2 Core Data Model, HCI Style Guide, Security
- JTA digital comm. standards not employed
  - Combat Net Radio (mandated), Wireless LAN (emerging)
- Legacy information technology
  - Assembly language, C, FORTRAN, JOVIAL
  - System-unique operating systems and run-time executives
  - ATDL-1, GBDL, Link-11 (TADIL A), Link-11B (TADIL B), PADIL, TRI-TAC (planning)
  - System-unique data models

SCENARIO #2 (OPERATIONAL CA. 2003)

The second scenario adds the THAAD system, along with upgrades to the earlier components. These systems continue to be mostly self-contained, stand-alone systems with "organic" sensors, interceptors, and battle management and command and control systems. These systems now contribute more directly to the common tactical picture through the Joint Data Network. There is still relatively little communication or interoperability horizontally between systems, however. Although such interactions are not part of current plans or system specifications, they play a major role in potential capability enhancements.

One new feature introduced in this scenario is cueing one TMD weapon system from another. Early warning sensors provide a conventional form of cueing. The THAAD radar, for example, can provide similar early notice of incoming missile tracks to an underlying PATRIOT system, or to a successor THAAD or AEGIS system down-range. A more advanced capability would not merely notify other systems of a threat but would negotiate a commitment from the best-positioned or best-able defender to engage the threat. This type of interaction would achieve dynamic system tasking and would make workload and resource balancing across TMD systems possible.

SCENARIO #2 (OPERATIONAL CA. 2003)

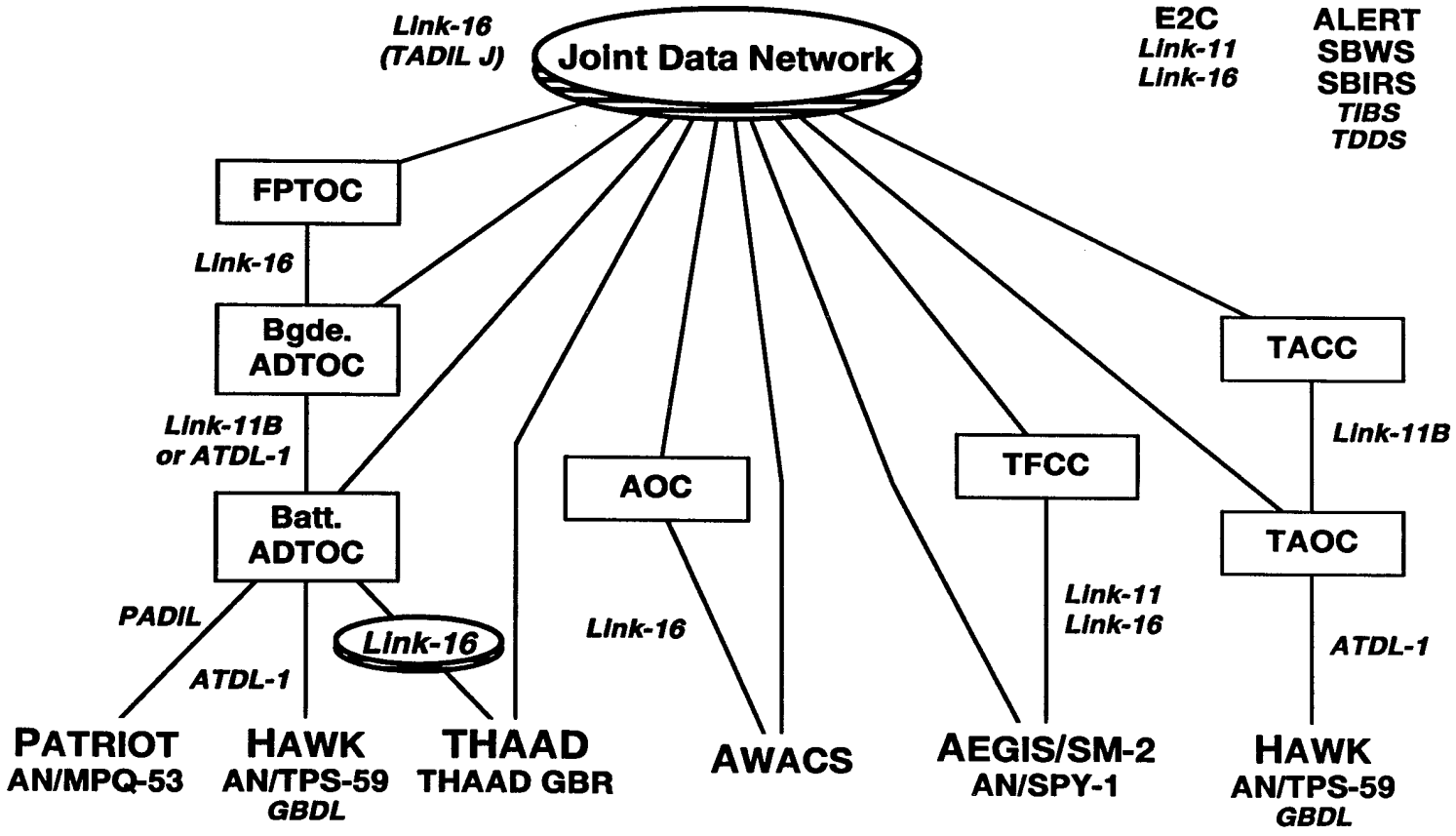
- AEGIS (NAD), HAWK, PATRIOT, THAAD
  - ALERT, SBWS, SBIRS warnings; AWACS and E2C track data
- Stand-alone systems, composite tactical picture
  - Uniquely coupled, closed systems (stovepipes)
  - Assignments, ROE, intelligence, early warnings passed down
  - Status reported up, tactical picture data reported directly to JDN
  - Underlay and successor tier cueing (like EW cueing)
  - Locally autonomous engagement decisions within constraints
  - No real-time battle coordination among systems
- Adaptability
  - Limited mix-and-match reconfiguration capabilities

COMMUNICATIONS FOR SCENARIO #2

This diagram shows evolving command and control systems and new missile defense systems connecting directly to the Joint Data Network, while legacy HAWK and PATRIOT missile systems continue to use outmoded communication channels and protocols.

The communications architecture in this diagram is flatter than in the previous scenario, reflecting some potential for more horizontal communications between systems. For example, PATRIOT battalions, THAAD batteries, and AEGIS ships all report status information through the Joint Data Network (JDN). It appears, however, that a single Link 16 network does not provide sufficient bandwidth to include direct horizontal communications among these batteries over the JDN. This is partly due to Link 16's time-division multiplexing of individual channels. This means that the JDN will continue to be used primarily for vertical communications.

COMMUNICATIONS FOR SCENARIO #2



IDA

STANDARDS FOR SCENARIO #2

This list of standards is identical to that for Scenario 1. Although there are proportionately fewer systems using the legacy communications technology, they are all still in active use. Evolution to more modern technology has not been achieved. This technology is now another five years older than in scenario number one, while the threats these systems might encounter are that much newer, faster, and more sophisticated.

Link 11B (TADIL B) is slated for retirement soon after the time frame for this scenario (ca. 2005).

STANDARDS FOR SCENARIO #2

- Standards in compliance with BMD standards profile
  - Ada, POSIX, Link-16 (TADIL J), MILSATCOM (TDDS, TIBS), C2 Core Data Model, HCI Style Guide, Security
- JTA digital comm. standards not employed
  - Combat Net Radio (mandated), Wireless LAN (emerging)
- Legacy information technology
  - Assembly language, C, FORTRAN, JOVIAL
  - System-unique operating systems and run-time executives
  - ATDL-1, GBDL, Link-11 (TADIL A), Link-11B (TADIL B), PADIL, TRI-TAC (planning)
  - System-unique data models

SCENARIO #3 (BEYOND 2003)

For the third scenario we have taken the same systems that appeared in the second scenario and added flexible, high bandwidth communications between all systems. This makes possible a number of mission enhancements. For example, underlay and successor tier cueing can now become a coordinated action where two (or more) systems combine resources to provide an improved defensive response. An extension of such system collaboration leads to a coordinated shoot-look-shoot capability, where the second shot is launched from a second system. A system that has a blind spot in one direction could use sensor data from another system to fill in track data it is missing. Separate batteries could "share" interceptors. A battery that picks up a track while its launcher is being serviced could have launch capabilities provided by a neighboring battery. The neighbor would launch an interceptor and hand control of fly-out back to the original tracking battery.

SCENARIO #3 (BEYOND 2003)

- Objective: "mix and match" system flexibility to enhance mission capabilities
- AEGIS (NAD), HAWK, PATRIOT, THAAD
  - ALERT, SBWS, SBIRS warnings; AWACS and E2C track data
- Sample mission enhancements
  - Coordinated underlay and successor tier tasking
  - Coordinated shoot-look-shoot across multiple batteries
  - Launch on track data from external sensor(s)
  - Interceptor hand-off to second battery
  - Open coupling of organic sensor-weapon combinations
  - Increased coordination at lower levels of command

SCENARIO #3 (CONTINUED)

The potential mission enhancement on the previous slide indicate the need for new communications messages and, in particular, new patterns of message exchanges among BMD systems. These include both computer to computer messages and human in the loop command and control messages.

These patterns of message exchanges, the decisions and actions they are associated with, their coordination, and their timing are called "protocols." Protocols listed here all have characteristics that are specific to BMD system functions and mission operations. They are not part of any existing message sets.

A drawback of the way military communications systems have been developed in the past has been a tight coupling between communications technology and the message sets they carry. Incompatibilities between technologies causes one level of difficulties; incompatibilities between message sets compounds the problem. This is contrast to a "layered" approach, which separates higher-level messages and protocols from lower-level protocols and standards which apply to specific mediums. The objective of layering is to allow any necessary message to be transmitted over any available carrier. One key to achieving communications flexibility is to separate the messages from the medium.

Another key is equitably sharing the available bandwidth. Conventional time-division multiplexing (TDM) techniques allocate fixed slices of bandwidth that go wasted when a participant has nothing to transmit. Static TDM provides no way for participants to gain more slots when the need arises and, therefore, handles bursts of information transfer poorly.

SCENARIO #3 (CONTINUED)

Protocols:
- patterns of messages
- decisions/actions
- acknowledgments/ coordination
- message acknowledgment and timing

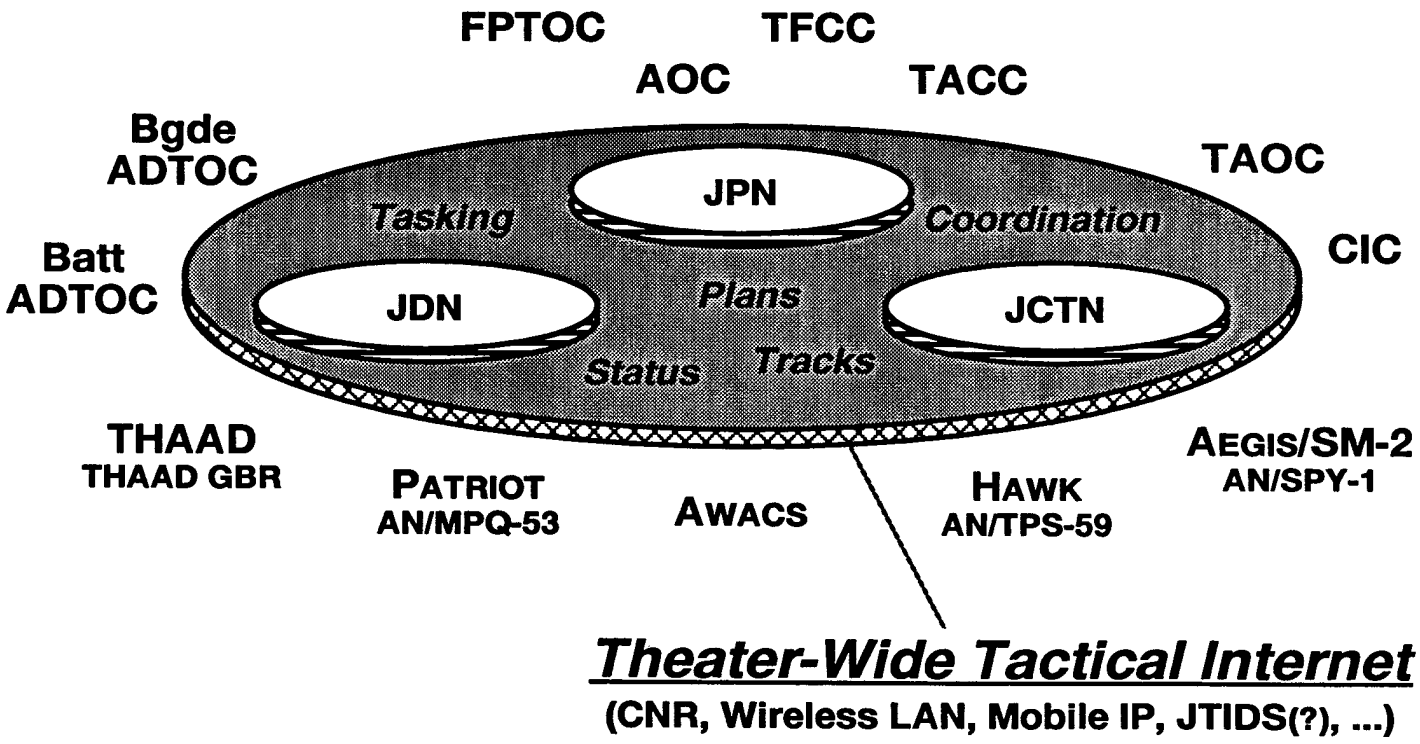
- BMD-specific patterns of message exchanges
  - Track Maintenance Protocol
  - Sensor Tasking Protocol
  - Underlay and Successor Tier Tasking Protocol
  - Coordinated Shoot-Look-Shoot Protocol
  - Interceptor Control Protocol
  - Composite Tracking Protocol
- Communications bandwidth and flexibility
  - Need more bandwidth
  - Need smart data distribution — dynamic bandwidth sharing
  - Need to decouple message sets from communications devices
  - Current tight coupling severely restricts flexibility
  - Solve bandwidth problem as a separate issue

COMMUNICATIONS FOR SCENARIO #3

This diagram attempts to portray theater communications as a large "flat" network with flexible interconnections. The TMD Joint Data Network, Joint Planning Network, and Joint Composite Tracking Network are shown as parts of the larger theater-wide network. Individual TMD components can plug into this network and establish any connections and transmit any messages necessary to support mission operations. Internet technology provides the desired level of flexibility today and, in the time-frame of this scenario, is expected to have evolved to meet TMD bandwidth and guaranteed quality of service needs.

All of the conventional vertical command and control communications are fully supported by this network. In addition, the network will support the real-time horizontal communications necessary for the enhanced BMD capabilities described above.

COMMUNICATIONS FOR SCENARIO #3



STANDARDS FOR SCENARIO #3

Standards necessary to support this scenario include network communications equipment standards, low-level network protocol standards, and higher-level command and control and BMD-specific protocols. The Global Command and Control System (GCCS) and Defense Information Infrastructure (DII) plan to evolve to DCE and CORBA for command and control transactions. This technology would allow BMD to construct flexible application-specific transactions to support its mission.

Standard protocols and layers for BMD-specific messages could serve a useful supporting role (see page 30 and page 76). Protocols for track maintenance, sensor tasking, and composite tracking are likely to apply to all of theater air defense, if not more widely. Underlay cueing, shoot-look-shoot, and interceptor control protocols may be specific to ballistic missile defense.

STANDARDS FOR SCENARIO #3

- Standard communication links, inter-networking protocols, bridges, gateways, etc.
  - Internet protocol suite
  - Battlefield Internet communications support (CNR?, ABIS?)
  - JTIDS role? (interoperability with IP?, bandwidth?)
- Standard distributed C2 transactions (vertical, non-real-time)
  - GCCS, DII
  - Possible evolution to DCE or CORBA — fault-tolerance and dynamic reconfiguration
  - Collaborative planning and coordination transactions
- BMD-specific transaction protocols (horizontal, real-time)
  - Track maintenance, sensor tasking, composite tracking
  - Underlay cueing, shoot-look-shoot, interceptor control

TMD SUMMARY

Collaboration among missile defense systems appears to have potential for enhancing overall mission effectiveness. Conventional command and control communications technology employed in current TMD system designs lacks the flexibility to achieve many of these enhancements.

Battlefield internet technology, which at present is a rapidly evolving DoD research and development area, would provide the communications bandwidth and connection flexibility needed. To capitalize on these advances, TMD systems must separate message information from the communications devices that carry the messages; migrate toward common open communications between sensors, weapons, and BMC3 components; and establish common transaction protocols for communications both within systems and between systems.

TMD SUMMARY

- Current planning assumes limited interoperability among TMD systems
  - Scenarios 1 and 2 show limited evolution toward interoperability standards, limited mix-and-match flexibility
  - Need plans for upgrading communication links
- BMD can leverage component system capabilities to achieve significant mission enhancements
  - Open uniquely-coupled sensor/weapon systems (stovepipes)
  - Establish BMD-specific transaction protocols (patterns of messages)
  - Separate message sets from communications devices (address information exchange and bandwidth problems separately)
  - Plan migration from Link 16 to battlefield Internet technology as this technology matures

NMD EVOLUTION SCENARIOS

In this series of slides the discussion moves from TMD to NMD scenarios. The goals are essentially the same as before. Because of present missile defense treaties and the increased scale of distances and areas involved in NMD versus TMD, there has been little discussion of communication between NMD and TMD systems. If NMD is built as a stand-alone system, as TMD systems have been, its ability to collaborate with TMD resources would be limited if the need were to arise. Although BMDO has no plans for joint NMD/TMD operations, this part of our study explored potential interconnections between NMD and TMD systems and the standards that would be needed to support such operations.

The Program Update Studies (PROGRUS) provided recommendations to BMDO on NMD and TMD architecture and technology strategies based on evolving assessments of threats and response capabilities.

The description of the NMD system here was developed in advance of (and independently of) the roadmap being prepared by the NMD System Engineering and Integration (SE&I) contract.

NMD EVOLUTION SCENARIOS

- Purpose and goals
  - Identify interoperability and adaptability needs — no system is a stand-alone system
  - Explore potential mission-enhancing interactions between NMD and TMD systems
  - Identify where existing standards could be applied and areas where new (BMD-specific) standards would be needed
- Basis for formulating scenarios
  - PROGRUS presentations
  - Speculation in advance of NMD SE&I roadmap

SCENARIO #4 (OPERATIONAL CA. 2003)

The structure of the NMD system is similar to individual TMD systems. The primary differences are in the scaled-up range of sensors and interceptors, scaled-up BMC3 capacity, and scaled-up bandwidth of communications between components. Because of NMD's unique mission, it would be easy for it to be built as a stand-alone system, with unique connections between its component subsystems and, other than conventional command and control messages, no abilities to exchange information with other systems in real time. Because of the global geometries involved in NMD, there may be few opportunities for significant system reconfigurations on-the-fly and, therefore, no requirements to build in such capabilities.

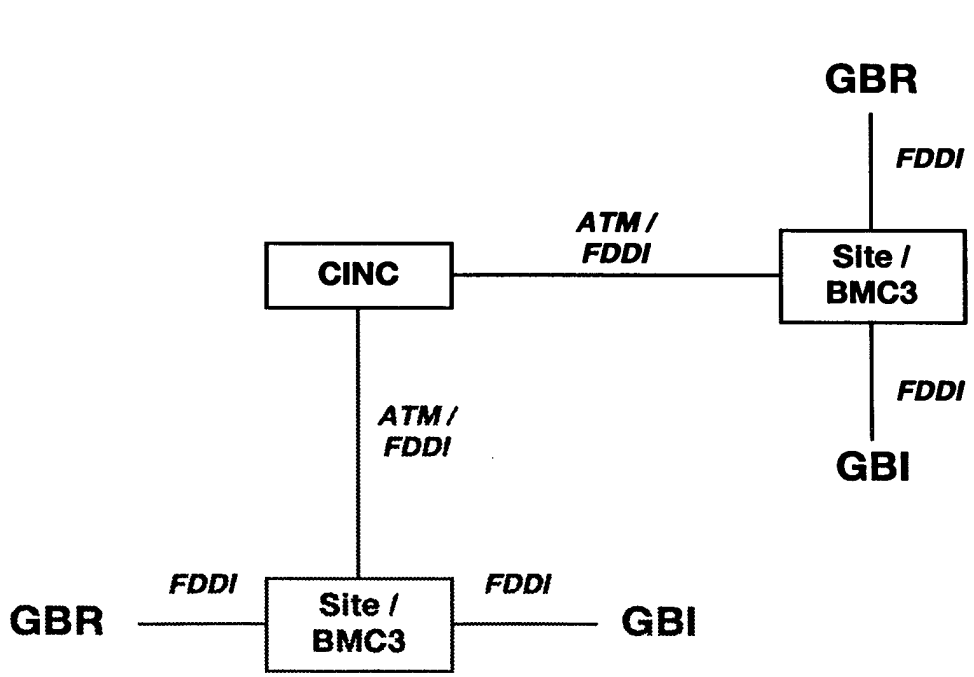
SCENARIO #4 (OPERATIONAL CA. 2003)

- Single- or multiple-site GBI, GBR, and BMC3
  - ALERT, SBWS, SBIRS warnings
- Stand-alone system
  - Uniquely coupled, closed subsystems (stovepipe)
  - Assignments, ROE, intelligence, early warnings passed down
  - Status reported up
  - Locally autonomous engagement decisions within constraints
  - No real-time battle coordination with any other systems
- Adaptability
  - Limited mix-and-match reconfiguration capabilities

COMMUNICATIONS FOR SCENARIO #4

Reliable, high-bandwidth communications between component systems appears to be a relatively straightforward exercise with commercial Asynchronous Transfer Mode (ATM) services and dedicated fiber-optic cable links. The sites of NMD sensors, interceptors, and BMC3 units will be positioned at large distances from the CINC and each other, making tactical radio links ineffective. Satellite communications could serve as a back-up to the land-based links.

COMMUNICATIONS FOR SCENARIO #4



ALERT
SBWS
SBIRS
TIBS
TDDS

IDA

STANDARDS FOR SCENARIO #4

Because of their development time-frame, all NMD component systems should be in close compliance with all current DoD and BMDO technical architecture mandates. While some of these systems may be derived from existing systems, all are expected to be upgraded to complete standards compliance in the process of achieving their NMD functional and performance requirements.

STANDARDS FOR SCENARIO #4

- Standards in compliance with BMD information technology standards profile
  - Ada, POSIX, ATM, FDDI, MILSATCOM (TDDS, TIBS), C2 Core Data Model, HCI Style Guide, Security
- Legacy information technology
  - Should be reasonably up to date for GBR and BMC3
  - For GBI, depends on basis for system development

SCENARIO #5 (BEYOND 2003)

This scenario looks at how TMD systems could be configured to augment NMD systems to defend against theater-range threats to US territory and resources. Part of the rationale here is that theater-level defenses should be more effective against theater-level threats than hemisphere-level defenses and less threatening to world order.

Standing up a THAAD or AEGIS defense for another Cuban missile crisis, for example, might provide a more effective defense and prove less destabilizing than exercising NMD resources

Our objective is not to promote deployment of such configurations, but to analyze what might prevent the interconnection of NMD and TMD resources if the need were to arise. If we look at TMD resources as a successor tier to NMD, all of the mission enhancements discussed earlier for TMD have counterparts in joint NMD/TMD operations.

SCENARIO #5 (BEYOND 2003)

- Single- or multiple-site GBI, GBR, and BMC3 augmented with TMD resources
  - AEGIS (NAD), HAWK, PATRIOT, THAAD
  - ALERT, SBWS, SBIRS warnings; AWACS and E2C track data
- Sample mission enhancements
  - Coordinated underlay (NMD to TMD) and successor tier (TMD to NMD) tasking
  - Coordinated shoot-look-shoot across NMD and TMD systems
  - NMD launch on track data from TMD sensor(s) and vice-versa
  - Interceptor hand-offs across NMD and TMD systems
  - Open coupling of organic sensor-weapon combinations
  - Increased coordination at lower levels of command

SCENARIO #5 (CONTINUED)

Since missile defense is missile defense, independent of the scale of the defended area, all the same types of information transactions are expected take place. One major difference between NMD and TMD systems, however, is their communications. TMD systems must be mobile, communicate over shorter distances, and carry their own radio equipment. NMD systems are fixed in place and can use fixed communications services. The advantage of decoupling message sets from communications devices shows up again here, as shown in the diagram showing communications for scenario #5.

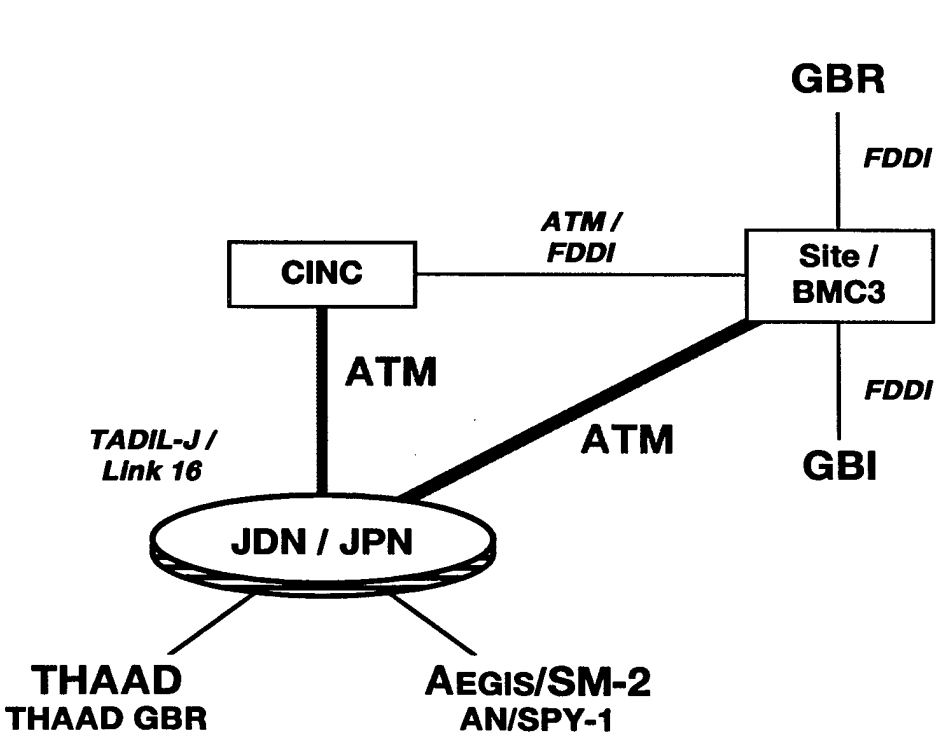
SCENARIO #5 (CONTINUED)

- BMD-specific patterns of message exchanges
  - Track Maintenance Protocol
  - Sensor Tasking Protocol
  - Underlay and Successor Tier Tasking Protocol
  - Coordinated Shoot-Look-Shoot Protocol
  - Interceptor Control Protocol
  - Composite Tracking Protocol
- Communications flexibility
  - Need to connect NMD and TMD communications
  - Need to decouple message sets from communications devices

COMMUNICATIONS FOR SCENARIO #5

With standard message protocols across NMD and TMD systems, connections between the two would require nothing more than equipment to connect TMD systems to an ATM service. The connection shown here is through either the Joint Data Network or the Joint Planning Network (or both). This allows local "theater" information to flow between theater systems while connecting the theater group to NMD information and command and control. In a later time frame the TMD systems would connect to a battlefield Internet and the ATM links to NMD would simply be an extension of that network.

COMMUNICATIONS FOR SCENARIO #5



ALERT
SBWS
SBIRS
TIBS
TDDS

IDA

STANDARDS FOR SCENARIO #5

The standards necessary for this scenario are the same as for the advanced TMD scenario, with a few additional requirements for bridging between JDN/JPN technology and high-speed NMD communications technology. This should be straightforward, provided that the message sets are separated from carrier technology, and NMD and TMD systems share common application-level transaction protocols. Protocols for track maintenance, sensor tasking, and composite tracking are likely to apply to all of air defense, if not more widely. Underlay cueing, shoot-look-shoot, and interceptor control protocols may be specific to ballistic missile defense.

STANDARDS FOR SCENARIO #5

- Standard communication links, internetworking protocols, bridges, gateways, etc.
  - Internet protocol suite
  - Battlefield internet communications support (CNR?, ABIS?)
  - JTIDS role? (interoperability with IP?, bandwidth?)
- Standard distributed C2 transactions (vertical, non-real-time)
  - GCCS, DII
  - Possible evolution to DCE or CORBA — fault-tolerance and dynamic reconfiguration
  - Collaborative planning and coordination transactions
- BMD-specific transaction protocols (horizontal, real-time)
  - Track maintenance, sensor tasking, composite tracking
  - Underlay cueing, shoot-look-shoot, interceptor control

SCENARIO ANALYSIS CONCLUSIONS

The preceding charts presented a time series of BMD system scenarios that serve to depict stages of mission capability evolution as well as potential system interactions important to mission capability. Although based partly on known programmatic data, they must be considered hypothetical because official and complete BMD evolution goals are not finally resolved.

In our view, evolution scenarios are needed for several reasons. They are valuable for an initial understanding of the compliance of currently planned programs with commercial standards such as those mandated in the DoD Joint Technical Architecture. Scenarios also exhibit where commercial standards may come into play in mission operations. Even more important, scenarios are an aid for identifying standards that are needed across BMD systems, but not yet recognized. They also may reveal other architectural issues that could impede interchange of weapon and sensor data across Service systems.

As a product of this analysis, we have identified the need for six BMD-specific protocols that are key to interoperability and achieving highly flexible missile defense (and air defense) operations. Standardizing and evolving these protocols is important to achieving open system characteristics for the BMD system of systems, e.g., a "plug and play" approach to integrating new or much upgraded element systems.

SCENARIO ANALYSIS CONCLUSIONS

- BMD or any mission domain needs planned evolution scenarios
  - Exhibit standards compliance in current and future plans
  - Indicate mission value and application of mandated standards
  - Aid recognizing other or mission-specific standards needed
  - Reveal architecture issues that could impede interoperability
- BMDO should consider defining open protocols for joint operations
  - Six candidates identified via scenario analysis
  - Key to highly flexible and adaptable system of systems
  - Open specification important to "plug and play" evolution and "mix and match" deployments

ANALYSIS OF BMD INTEROPERABILITY STANDARDS

The following charts and annotations are an analysis of selected Ballistic Missile Defense interoperability standards.

ANALYSIS OF BMD INTEROPERABILITY STANDARDS

PURPOSE AND OUTLINE

The purpose of this section is to analyze selected current and potential BMD interoperability standards to identify needs and required actions by BMDO. The previous charts showing BMD scenarios are a "top down" view of the BMD system of systems showing where standards are needed. The following charts are a "bottom up" analysis of existing standards and sets of standards, complementing the scenario analysis. These standards were analyzed to determine their weaknesses (e.g. support for real-time systems) compared to BMDO needs.

The Joint Technical Architecture (JTA) was examined to identify key interoperability standards and the limitations of the Joint TA. The Defense Information Infrastructure (DII) and Global Command and Control System (GCCS) Common Operating Environment (COE) were also examined to identify key standards and limitations. Selected other standards areas were also examined.

PURPOSE AND OUTLINE

Purpose of section: Analyze selected current and potential BMD interoperability standards to identify needs (e.g. real-time support) and required action

- **JTA standards and limitations**
- **DII and GCCS COE standards and limitations**
- **Selected standards areas**
  - **Application-level communications: DCE, CORBA, COM**
  - **Lower-level communications: Link-16 and commercial approaches (TCP/IP, ATM)**
  - **Computer/Information Security**

JOINT TECHNICAL ARCHITECTURE

The Joint Technical Architecture (JTA) version 1.0 was mandated on August 22, 1996. Its current scope is DoD Command, Control, Communications, Computers, and Intelligence (C4I), though the intent is to widen that scope in the future [DoD 1996, 2].

The JTA mandates a set of standards in the sense that, if a system needs a service area identified in the JTA, the system must use the standards identified in the JTA [DoD 1996, 1-6]. Sample key standards are listed here.

Note that the JTA mandates that C4I systems be at least DII COE level 5 compliant, with a goal of level 8 compliance. The DII COE and these compliance levels are discussed in the following charts.

A limitation of the JTA is that more than one standard may be allowed to implement a given service or capability, permitting incompatibility. For example, both CORBA and DCE can be used to support inter-application communication. It is quite plausible for two different systems to each use one and make it difficult to support inter-application communication across the two systems. The JTA also does not identify how standards should be used together. Perhaps most important, the JTA does not contain all the services and standards that BMD will need to achieve complete interoperability and portability.

JOINT TECHNICAL ARCHITECTURE

- Version 1.0 mandated for DoD C4I (scope to be broadened); if a service is used, it must be one of the mandated standards
- Mandated standards include:
  - Information Processing: DII COE level 5 (goal 8), POSIX/CDE (Motif and X-windows) or Win32, SQL, ODBC, DCE, CORBA
  - Information Transfer: TCP/IP, ATM, SONET, JTIDS medium
  - Information Modeling: IDEF0/1X, DDDS, Link 16/Link 22/VMF
  - Human-Computer Interfaces: DoD HCI Style Guide, Motif/Win32
  - Security: Fortezza, various policies; many emerging standards
- Limitations:
  - Often more than one standard can be used to implement an approach, permitting incompatibility (e.g. CORBA vs. DCE)
  - Does not identify how to use standards together
  - Does not contain all services/standards that BMD would need

GCCS AND DII COE BACKGROUND

In September 1996 the Global Command and Control System (GCCS) replaced the older World-Wide Military Command and Control System (WWMCCS). GCCS is to implement the "C4I for the Warrior" (C4IFW) concept, leverage commercial off-the-shelf software, and be portable to a wide range of system architectures [MNS 1995]. The GCCS project developed a Common Operating Environment (COE), which has since been generalized into the Defense Information Infrastructure (DII) COE. GCCS is now one of the users of the DII COE.

The DII COE packages software and databases into "segments," simplifying installation, upgrading, and CM. This segmentation construct is a key feature of the DII COE, and the tools to support segmentation are available now. As discussed in the next chart, applications cannot reach level 4 or higher levels of compliance without using segmentation.

The DII COE is to provide a number of standardized COE segments, which when available could be beneficial to BMDO. Unfortunately, the objective requirements and APIs for nearly all such segments are not yet available.

The next two charts present in more detail what is meant by "DII COE compliance" and what standards or products are included in the DII COE.

GCCS AND DII COE BACKGROUND

- GCCS has replaced WWMCCS
- GCCS is to implement C4IFW concept and use DII COE
- DII COE:
  - Packages software and databases into "segments," simplifying installation, upgrading, and CM. Available now
  - Will provide standardized COE segments: support applications (alerts, correlation, MCG&I, etc.) and infrastructure segments. These could lower development costs and simplify integration
  - Runs on Sun Solaris and HP HP-UX; to be partially available on Windows NT; porting ongoing for IBM AIX, DEC Alpha, SGI
  - DCE/CORBA use in debate. DISA's CORBA strategy document says DCE will be used and eventually CORBA on top of DCE; this is to be seen
  - Objective requirements and APIs for COE segments are not available—they are in various stages of development

DII COE COMPLIANCE

The DII COE Integration and Run-Time Specification (I&RTS) defines four compliance categories. The "runtime environment" category has received the most interest because 8 levels are explicitly defined for this category and level 5 of this category is mandated by the JTA. At this time no explicit levels are defined for the other compliance categories, so these other categories are not discussed further here.

Higher levels in the runtime environment compliance category indicate increased compliance by an application. Level 1 is the lowest, simply requiring that a given application use COTS standards in the DII COE (e.g. POSIX/CDE or Microsoft Windows and SQL databases); level 3 requires that the application not interfere with other applications.

Level 4 of the runtime compliance category is the first level to require specific application changes to support the DII COE by requiring that the application be broken into segments. Level 5 requires that the application be broken into segments so it can be installed, uninstalled, and executed in the normal way by its users. This requires more detailed information about the application to be included with the segment(s).

The higher levels imply increasing amounts of reuse of other COE segments to reduce duplication, and increasing use of published interfaces to eliminate dependence on proprietary interfaces. For level 8, only public and published application programmer interfaces (API) may be used. In addition, to obtain level 8 compliance no other segment's functionality may be duplicated and the user interface style guide must be followed. [I&RTS 1995, 2-15].

Since level 8 requires no duplication, it's important to understand what is already in the DII COE; this is the subject of the next chart.

DII COE COMPLIANCE

- **Four compliance categories: Runtime Environment; Style Guide ("look and feel"); Architectural Compatibility (e.g. uses DCE where appropriate); Software Quality (software metrics)**
- **Runtime Environment has 8 compliance levels (level 5 mandated):**
  1. **Common set of COTS standards**
  2. **Co-exist on same LAN (different CPUs)**
  3. **Co-exist on same workstation**
  4. **All applications segmented and share the bootstrap COE**
  5. **Functionality accessed via Executive Manager; segments have more detail (boot, background, local processes, registration) and can be installed and removed using COE installation tools**
  6. **Use existing accounting groups and reuse 1+ COE segments**
  7. **Segments reuse (not duplicate) COE components; few if any private APIs are used**
  8. **No duplication of any segment, only public APIs, style guide**

Source: DII COE I&RTS

DII COE STANDARDS AND LIMITATIONS

Version 2.0 of the DII COE includes a set of COTS components, most of which are commonly-used components on Unix platforms.

The DII COE includes for Unix-based platforms the Common Desktop Environment (CDE), Motif, and X-windows for a graphical user interface (GUI). The Microsoft Windows NT version of the DII COE does not include this GUI; instead, it is expected that software would be written to use the GUI system built into Windows NT. TransArc's version of DCE is included in DII COE version 2.0. The current plan is to add CORBA in version 4.0 of the DII COE.

One military-specific component is the Joint Mapping Toolkit (JMTK) for Mapping, Charting, Geodesy & Intelligence (MCG&I).

The Office Automation Working Group has recommended that the Microsoft Office products be used for typical office tasks (word processing, etc.) to ease exchange of this sort of information. Version 2.0 of the DII COE included the Windows Application Binary Interface (WABI) to facilitate this.

There is only one source for maintenance of the DII COE (DISA). This could be a source of difficulty for BMDO, since the DII COE maintainer might not be able to make changes or additions (if necessary) in time for BMDO needs.

DII COE STANDARDS AND LIMITATIONS

- **Standards or Components (DII COE version 2.0):**
  - **CDE/Motif/X-windows (unless Windows NT)**
  - **Various COE segment management tools**
  - **DCE: TransArc (CORBA to be added)**
  - **SQL RDBMSs: Oracle, Sybase**
  - **Mapping, Charting, Geodesy & Intelligence (MCG&I): Joint Mapping Toolkit (JMTK)**
  - **Netscape's web browser and web server**
  - **WABI (to run Microsoft Office on Unix machines)**
- **Limitations:**
  - **Moving target: APIs and their requirements are in development**
  - **Few military-related support applications yet available**
  - **Single source for DII COE maintenance**

Source: DII COE 2.0 Baseline Specifications, Office Automation WG minutes

SELECTED STANDARDS AREAS

We also examined specific standards areas that directly involve interoperability and for which there is a lack of general consensus on a solution. These areas are further amplified in the following charts.

SELECTED STANDARDS AREAS

- Examined specific standards areas that:
  - Directly involve interoperability
  - Lack general consensus on a solution
- Areas are:
  - Application-level communications: DCE, CORBA, COM
  - Lower-level communications: Link-16 and commercial approaches (TCP/IP, ATM)
  - Computer/Information Security

APPLICATION-LEVEL COMMUNICATIONS: DCE, CORBA, AND COM

The Distributed Computing Environment (DCE), Common Object Request Broker Architecture (CORBA), and Component Object Model (COM) all attempt to simplify the development and deployment of distributed applications. Distributed COM (DCOM) is Microsoft's approach to distributed computing.

Each approach has its advantages. DCE includes a number of security mechanisms and is more mature; CORBA has stronger support for object-orientation and a variety of computer languages; DCOM has the financial support of Microsoft.

Simply selecting any of these mechanisms is insufficient. All of these mechanisms include "interface definition languages," therefore, specific interfaces must be defined before these mechanisms can be used and before any interoperability can be achieved using these mechanisms.

Additional information is included in Appendix A.

APPLICATION-LEVEL COMMUNICATIONS: DCE, CORBA, AND COM

- Both DCE and CORBA support distributed computing by automatically generating message formats, converting application data to and from messages, and exchanging messages
- DCE:
  - Based on remote procedure call (program-to-program)
  - Includes security features, time synchronization, directory
  - Other services can be built on DCE, e.g., transactions, CORBA
- CORBA:
  - Based on object-orientation (object-to-object)
  - Stronger support for non-C/C++ (Ada, Smalltalk, Java)
  - Specification supports asynchrony and dynamic invocation
  - Less mature than DCE
- COM/DCOM: Microsoft's approach
- Selecting one is not enough; interfaces using them must be defined

LOWER-LEVEL COMMUNICATIONS: LINK-16 BACKGROUND

Before contrasting military and commercial approaches to lower-level communication, it is necessary to review them. We will start with a review of link-16.

Link-16/TADIL J is a combined communications media, protocol, and message set developed in the 1970s for tactical communication. Link-16 can transmit data at an instantaneous rate of 28.8 kilobits per second (kbps), 57.6kbps, or 115.2kbps (at decreasing levels of jam resistance). It is limited to line-of-sight unless relayed.

Link-16 has a number of strengths. For example, link-16 was designed to be jam-resistant (through the use of mechanisms such as frequency hopping) and private (through encryption). Its message set defines a number of joint agreements on tactical messages. An analysis of link-16 for TMD uses is contained in the TMD C2 plan [BMDO 1996]; that analysis found that link-16 is sufficient for TMD needs. That analysis noted that many new users of link-16 have materialized, and that unless efficiency-enhancing techniques are imminent, air defense and TMD performance, especially in large-scale networks, will suffer [BMDO 1996, I-6]. Note also that the analysis assumed that TMD traffic would have priority over air track messages, which could lengthen the air track update interval [BMDO 1996, I-A1-28].

However, link-16 has disadvantages as well. Note that link-16 requires significant preconfiguration and setup for use. This is in part due to the static preallocation of time slots; each second is divided into 128 time slots, and each link-16 user on a network (frequency hopping pattern) is assigned specific time slots. If users transmit simultaneously on the same frequency (network), at best others will hear the closest sender. Dynamic time slot reallocation (TSR) is intended for the future to improve efficiency, but it is expected that many users will never have this capability.

The amount of bandwidth available is dependent on a very large number of factors. Since there are 128 time slots, one simple estimation approach is assume that the networks is nearly optimally preallocated and few overhead messages are transmitted; this results in 100 time slots/second available in wartime. If strong jam resistance is needed, this amounts to 100 messages/second (or 20 messages/second for 5 sensors perfectly sharing the network). Note that this bandwidth is only available in wartime; peacetime power restrictions require that the entire network only be used 40% of the time, with any one terminal using at most 20% of the time. These can be overridden in exercises, but care must be taken since it can potentially interfere with national air navigation and flight safety [Logicon 1994, 5-63] [Golliday 1985].

In military approaches the message formats are strongly tied to the specific frequency and bandwidth, as opposed to the commercial approaches that are discussed next.

LOWER-LEVEL COMMUNICATIONS: LINK-16 BACKGROUND

- Link-16/TADIL J:
  - Developed in 1970s to exchange real-time tactical data
  - UHF Lx radio link: line-of-sight (unless relayed) at 28.8-115kbps
- Strengths:
  - Security: jam-resistant (freq. hopping) and private (encryption)
  - Defined set of tactical messages
  - Official primary DoD tactical data link (link-22 and VMF OK too)
  - Sufficient though limiting for TMD (Source: TMD C2 Plan)
- Weaknesses:
  - Current JTIDS terminals expensive; MIDS to help
  - Requires significant preconfiguration and setup for use, currently in part due to static preallocation (128 time slots/sec)
  - Bandwidth limited: each network handles ~100 track reports at 1s rate; less in peacetime (40%/20%). 5 sensors $\Rightarrow$ ~20 tracks/s
  - Tactical messages tied to specific radio type (monolithic)

LOWER-LEVEL COMMUNICATIONS: COMMERCIAL APPROACHES BACKGROUND

Commercial approaches separate media, protocols, and messages from each other, in contrast to the approach taken by link-16. For example, commercial software does not need to be rewritten when higher-bandwidth modems are available. Thus, the emphasis in the commercial world has been on communication protocols which can be carried over a variety of media.

The TCP/IP protocol suite used by the Internet is extremely popular due to its ability to connect systems with very different underlying mediums. However, TCP/IP has a number of weaknesses that are important to the military. It does not handle streaming audio or video data well. It does not support mechanisms to define guaranteed quality of service. These weaknesses are being addressed; extensions to support streaming and quality of service have cleared the drafting committees. While some small attempts were made to include security features, TCP/IP does not include sufficient features by itself to permit secure transmissions. TCP/IP also includes overhead which makes it less suitable for very low bandwidth media. TCP/IP transmits data in "packets" with a header, and the TCP/IP header is 224 to 320 bits long (5 words for IP, plus 2 words for UDP or 5 words for TCP, assuming no options are used; each word is 32 bits long) [Comer 1991, Postel 1980, Postel 1981a, Postel 1981b]. This header is a significant overhead on low bandwidth media, although there have been some proposals that attempt to mitigate this [Jacobson 1990].

Asynchronous Transfer Mode (ATM) is another protocol, based on small, fixed-size packets. It is designed to carry voice, data, and video, and supports different levels of quality of service [Goralski 1995].

Ongoing military-related work is attempting to take advantage of commercial technology for military purposes. These include Global Broadcast Service/Direct Broadcast Service (GBS/DBS) to provide high speed data dissemination to the warrior [DDR&E 1995]. MIL-STD-188-220A (27 July 1995) defines a mechanism for using TCP/IP over combat net radio (CNR).

LOWER-LEVEL COMMUNICATIONS: COMMERCIAL APPROACHES BACKGROUND

- Commercial approaches separate media, protocols, and messages
  - Improves flexibility by permitting the use of different bandwidths and mediums; all support dynamic reallocation
- Internet (TCP/IP) Protocols
  - General internetwork computer communications approach with vast commercial support
  - Does not currently handle well: streaming (audio/video), quality of service, security, very low bandwidth
- Asynchronous Transfer Mode (ATM) Protocol
  - Designed to carry voice, data, and video using small packets
  - Supports different levels of quality of service
  - Newer, does not currently handle security
- Ongoing military-related work: Global Grid, DBS/GBS, CNR
- Many mediums supported, especially fiber optic and wire

LOWER-LEVEL COMMUNICATIONS: LINK 16 AND COMMERCIAL APPROACHES

Clearly, each approach has strengths. Ideally, it would be nice to merge the strengths of the military and commercial approaches.

If tactical communication layers and interfaces were defined, the media, protocols, and messages could be separated in the military communication approaches. This would permit the use of different media (radio, fiber, wire) and bandwidth without changing application software. This is in contrast to current military approaches, which require application changes or the use of information-losing converters to support a change in bandwidth. For example, an existing system using link-16 requires application program changes or a conversion system to take advantage of link-22. By separating the communication layers, flexibility could be increased.

LOWER-LEVEL COMMUNICATIONS: LINK 16 AND COMMERCIAL APPROACHES

- Each approach has strengths:
  - Link-16 is jam-resistant and private, defines a tactical message set (implying many joint agreements), uses messages designed to fit in the radio bandwidth, and certain quality of service issues are guaranteed (due to preallocation of time slots)
  - Commercial approaches support flexibility, including non-radio media (important for NMD) and dynamic use of bandwidth
- Tactical communication layers and interfaces could be defined, separating media, protocols, and messages:
  - Permit use of different media (radio, fiber, wire) and bandwidth without changing application software
  - Could permit use of "best-fit" message formats for a given situation (link-16, link-22, VMF)
  - Would increase flexibility

The term "security" is a broad term covering a number of important requirements; as a result there are many definitions of security. One common definition is that security encompasses confidentiality, integrity, and availability; DoD requirements have in the past often concentrated on confidentiality [NRC 1991]. The security "common criteria" defines the following security functional requirements: audit, communication (i.e. non-repudiation), data protection, identification and authorization, protection of system, resource utilization, access, and trusted paths [CCP 1996].

Security is not a simple binary state (secure or insecure)—there are different degrees of security. Additional money and time can be spent to improve security, but it is impossible to have "perfect security" that cannot be broken. Instead, the degree of security needed depends on the threat (e.g., the cost of system compromise and the resources of the attacker). In the case of BMD, the cost of compromise is great (lives may be lost) and the attacker may have significant resources available.

If attackers will be limited to only attack the communication paths between BMD systems, simple countermeasures can aid significantly. JTIDS includes data encryption and checksumming measures to provide significant integrity and confidentiality of data, and JTIDS' frequency hopping and jam-resistance increase availability. Fiber lines can be bulk encrypted using NSA-approved equipment; combined with checksumming, this can provide some integrity and confidentiality. By itself this does not provide availability (the lines can be cut), so alternative transmission routes would be needed for availability.

However, protecting communication paths by no means handles all security issues. With significant resources an attacker can attack the systems (endpoints), attempt "insider" threats, and use electromagnetic emissions. Multi-level security may be necessary; while many BMD systems can run at system-high Secret, the intelligence systems they depend on often cannot.

There are many security-related or security-including interfaces. A few are listed here. Some standards include security capabilities, for example, JTIDS. Note that some standards and products can create security vulnerabilities (e.g. by incorrectly handling errors or performing requests without regard to the legitimacy of the request).

Security is an extremely broad topic. There are a number of standards which affect security in some way, but attempting to analyze all permutations of them is a futile exercise. Instead, BMDO must define its system security requirements and security threats, and show that they will be sufficiently met by its system architecture and supporting standards. This might be implemented through a "security system architecture" which would specify how various components must be combined to produce an acceptably secure result.

COMPUTER/INFORMATION SECURITY

- Security is multi-faceted (e.g. confidentiality, integrity, availability, audit, non-repudiation, authentication)
- Security has many degrees; correct degree based on the threat
- Many security-related or security-including interfaces, e.g.:
  - Security-related: Generic Security Service (GSS) API, FORTEZZA interface, Microsoft Crypto API
  - Security-including: JTIDS, DCE, eventually CORBA
  - Many standards and products have security vulnerabilities
- BMDO must define its system security requirements and security threats, and show that they will be sufficiently met by its system architecture and supporting standards.

INTEROPERABILITY PLANNING RECOMMENDATIONS

The following charts summarize recommendations emerging from the foregoing analyses, given that DoD and BMDO expect to continue efforts on Technical Architecture. Although interoperability remains the primary goal, the recommended efforts should begin to address adaptability for specific mission and commander requirements as well as lifecycle cost, an important driver of the DoD open systems initiative.

INTEROPERABILITY PLANNING RECOMMENDATIONS

STANDARDS SELECTION AND TRACKING

Rapid and rather unpredictable evolution continues to be a pronounced characteristic of information system technology and standards. Standardization has become an important factor in market definition and product development for open systems. The standards process need not lead to market acceptance and product maturity, so considerable caution is warranted before adopting newer standards. Consequently, moving toward JTA compliance as well as refining JTA content, has to be done with evolving technology and maturing market/product acceptance much in mind. Today's hot standards and products may become tomorrow's has-beens. Currently, JTA exhibits a tendency to add multiple alternatives instead of addressing the usability and maturity issues of individual standards.

Software standards of primary concern in JTA are distributed processing (DCE, CORBA) and mission support applications of the DII COE. Effort should be made to accumulate reported industry and DoD application experiences and problems, and develop well focused guidance on the challenges BMDO would face in applying them in BMC3. Several relevant prototyping and evaluation efforts have been reported and more are underway, including efforts at JNTF and IDA. It would help BMD developers and JTA participants to have the results summarized and analyzed for current usability conclusions. Also, BMDO may find from those results a need to undertake BMD-specific evaluation and prototyping studies.

A number of DoD initiatives or studies such as ABIS, Global Broadcast System, Joint Task Force Advanced Technology Demonstration, and battlefield Internet are providing results that may converge to make pertinent JTA standards compelling choices or add major new standards and influence toward particular applications of existing standards. BMDO should monitor and assess these situations and determine whether and how to add appropriate constraints into BMD evolutionary plans.

BMDO also should investigate potentially incomplete JTA areas, e.g., computer security, and investigate avenues for helping to accelerate progress on standards that may be needed. The latter includes standard application programming interfaces (APIs) for DII COE common support applications, battlefield Internet quality of service needed for BMD application, and mobility support for Combat Net Radio and any other tactical Internet capabilities.

STANDARDS SELECTION AND TRACKING

- Ongoing JTA updating should refine standards selection & advice
  - Some may be incomplete, e.g., security standards
  - Others conflict; performance/usability data are needed
- Significant effort is needed to monitor maturity and usability of JTA standards that will affect BMC3 software architecture
  - DCE, CORBA, DII COE products and usage experiences
  - Leverage DoD prototyping/evaluations, perhaps augment
- Significant effort is needed to monitor emerging technology alternatives that may affect future BMD system architectures
  - DoD battlefield Internet and other high-bandwidth comm
  - ABIS, JTF ATD, and other software-intensive capabilities
- Key standards and products needed for BMD mission in long-term need accelerating action, e.g., security APIs, COE APIs, IP quality of service, CNR/IP mobility

INTEROPERABILITY ARCHITECTURE DEVELOPMENT

Despite the value DoD recognizes in JTA Version 1, it is only an initial step toward the content envisioned for a technical architecture.* In our view, the BMD "system of systems" demands a more definitive, interoperability-focused technical architecture, or for brevity, an interoperability architecture. Such a TA would identify the common services or building blocks for flexible, mission-driven interoperation of the component systems. It also would provide a structure for identifying the commercial and mission-specific standards (and their options and parameters) to be used for meeting BMD functional and performance requirements. One step toward this goal would be to define a generic architecture or Technical Reference Model (TRM) for BMD systems. The TAFIM TRM [DISA 94] is overly generic, not technologically current, and inadequate for weapon and sensor concerns. It will not serve BMD's TA need.

IDA's project staff believes that BMD programs have not yet defined the set of "open system protocols" that would provide BMC3 interoperability for BMD future missions. Initial effort has been made in the limited JTIDS message extensions for sensor cueing and in ongoing effort on the Joint Composite Tracking Network. A larger effort is recommended to investigate a complete set of protocols for various types and levels of mission cooperation among BMD element systems. Open protocols should separate message types and content from the communications transport media that may carry them. Additional flexibility should be considered in the event that choices among alternative media are available.

Another JTA limitation that needs to be addressed for BMD is lack of standards for mission-oriented requirements. The problem becomes clear by considering the High Level Architecture for Modeling and Simulation (available at World Wide Web site www.dmso.gov) as well as the GCCS (or DII) COE services such as Alerts, Correlation, and Common Tactical Picture. BMD architectures should be structured so that they can reuse future COE segments if satisfactory for mission needs. The HLA for now is not part of JTA, and the form of future criteria to integrate HLA and JTA (for C4I) has not been investigated.

The impact of the COE and JTA needs to be made evident as BMD system architectures are modified or formulated for leverage and compliance. Explicit guidance to BMD programs about improving leverage and representing where standards will be applied will help future compliance and program assessments.

*The C4ISR Framework report [CISA 96] and the JTA's formal definition of technical architecture form the OSD guidance now available. Both indicate that a TA would be more than a list of preferred standards. They imply, as does the often-used analogy of "building codes", that a TA at least would clearly delimit the intended application of cited standards within a given mission domain such as National Missile Defense or Theater Air Defense.

INTEROPERABILITY ARCHITECTURE DEVELOPMENT

- Extend BMD Technical Architecture toward its expected scope
  - Formulate a TRM for BMD systems
  - Encourage JTA effort to improve on the TAFIM TRM
  - Include open system criteria in the TRM
- Initiate effort to define BMD open system protocols
  - Separate message content from comm medium
  - Improve flexibility in using other comm media as available
- Address standards for BMD mission-oriented applications
  - Require BMD architectures to improve potential reuse of COE
  - Investigate future criteria to integrate HLA and JTA in BMC3
- Require BMD architectures to specify where JTA standards will be applied

COMPLIANCE AND MIGRATION PLANNING

Achieving compliance of BMD systems with the Joint Technical Architecture will not be straightforward. Many disparate but important factors will come into play, including program cost and schedule impact. Many ambiguities and uncertainties also are involved in how DoD guidance will evolve. JTA version 1.0 deferred choosing standards for a number of services. Resolving those deferred areas and handling future technology and market changes will force further JTA updates while DoD programs are working to comply with prior releases. For cost and legacy reasons, achieving BMD compliance should not be attempted as a one-time event. Rather, it should be done as a series of actions or evolutionary stages, many coinciding with planned mission-driven BMD system upgrades or new acquisitions. Such a progression is called migration toward compliance.

BMDO will need to develop criteria for guiding migration decisions over the long-term. Migration for individual programs should not be decided entirely on short-term factors nor ignore migration plans for other interoperating systems. Guiding criteria should favor actions that will directly improve joint BMD capabilities and perhaps disfavor compliance to specific immature or emerging standards, at least until some promising usage condition or event occurs. Hence migration planning criteria may need support from a rating scheme or criteria that address the merits and difficulties of JTA standards as technology and market-driving artifacts, independent of their potential benefits for BMD applications and systems.

With interoperability so prominent in DoD concerns, coordinated action toward JTA compliance will be important. BMDO will need to identify programs where it has leadership responsibility for migration decisions, versus other programs where BMDO has some supporting role. Three categories may apply. One is BMD programs (of any acquisition size) where BMDO, possibly with a Service Executing Agent, has complete acquisition responsibility. A second is programs for systems that will interface to and support BMD operations and planning. but because of collateral missions are not designated as BMDO programs. Sensor systems may be good examples of this second category. A third would be programs that provide capabilities to be integrated into BMD operations, but have many uses in defense systems other than BMD. Examples of this third group include communications terminals and GPS receivers.

Coordination and cooperation also should prove valuable for engineering activities that support compliance decision-making. Steps to share information and avoid duplicative efforts such as similar prototyping or evaluation projects should be taken.

Compliance will involve making program assessments to judge progress. Assessment techniques need to be set up for constructive benefit, not simply to generate scores. For example, benefits could include identifying and sharing lessons learned in working toward compliance for some COTS standard product.

COMPLIANCE AND MIGRATION PLANNING

- Mandated standards will change as compliance effort proceeds
- Compliance needs an evolutionary or migration approach
- Migration planning should be driven by long-term criteria
  - Criteria should favor compliance for joint BMD capabilities
  - Rating JTA standards on maturity & performance may help
- BMDO and Services need coordinated BMD compliance efforts
  - Joint decisions on compliance action for BMD programs
  - Close coordination on Services' BMD-supporting systems
  - Sustained cooperation on BMD-related programs
  - Avoid duplication of standards tracking & prototyping efforts
- Compliance progress assessment should emphasize positive benefits such as sharing lessons learned on achieving compliance

APPENDIX A:
HIGH-LEVEL APPLICATION COMMUNICATION MECHANISMS: DCE, CORBA, AND COM

This appendix provides more detailed information on DCE, CORBA, and Microsoft's COM.

APPENDIX A:
HIGH-LEVEL APPLICATION COMMUNICATION MECHANISMS: DCE, CORBA, AND COM

APPENDIX A: OUTLINE

This appendix describes three approaches to high-level application communication. It first describes the problems these mechanisms attempt to solve and the general approach they all use. Each mechanism is then covered in the following order: DCE, CORBA, and COM. The mechanisms are then compared to each other and the important issue of CORBA immaturity is discussed. BMDO issues and a summary conclude this appendix.

APPENDIX A: OUTLINE

- Why DCE, CORBA, or COM?
- General Approach
- DCE
  - Basic Capabilities and Architecture
  - Backer and Products
  - Future Plans
- CORBA
- COM
- Issues: Comparison/Contrast, CORBA Immaturity
- BMDO Issues
- Summary

WHY DCE, CORBA, OR COM?

The goal of all of these mechanisms is to permit easy development and administration of distributed applications (communicating software on different computers). Distributed applications can be developed without these mechanisms, but programmers find that they need to solve a number of detailed problems for each application. For example, when applications communicate between different types of computers, the ordering of the bits in numbers and the order and arrangement of bits in a message must be agreed to. Various protocols to handle errors and to locate the other applications must be defined. Many other problems (such as security) often need to be solved as well, and it makes sense to use a single, common solution so that these problems do not need to be repeatedly solved for each application.

DCE and CORBA support distributed computing by automatically generating message formats, converting application data to and from messages, and exchanging messages. Distributed COM also supports inter-application communication.

WHY DCE, CORBA, OR COM?

- Goal: Easy development and administration of distributed applications (communicating software on different computers)
- Problems:
  - Data ordering (1234? 4321?) and packaging (meaning of bits)
  - Data protocols (acknowledgments, ignoring duplicates)
  - Finding the "other program," even when it moves
  - Many other capabilities (e.g. security) are often needed; common solution best
- Defining message formats solves the data ordering and packaging problem, but this is a tedious manual process
- DCE and CORBA support distributed computing by automatically generating message formats, converting application data to and from messages, and exchanging messages. Distributed COM also supports inter-application communication

GENERAL APPROACH

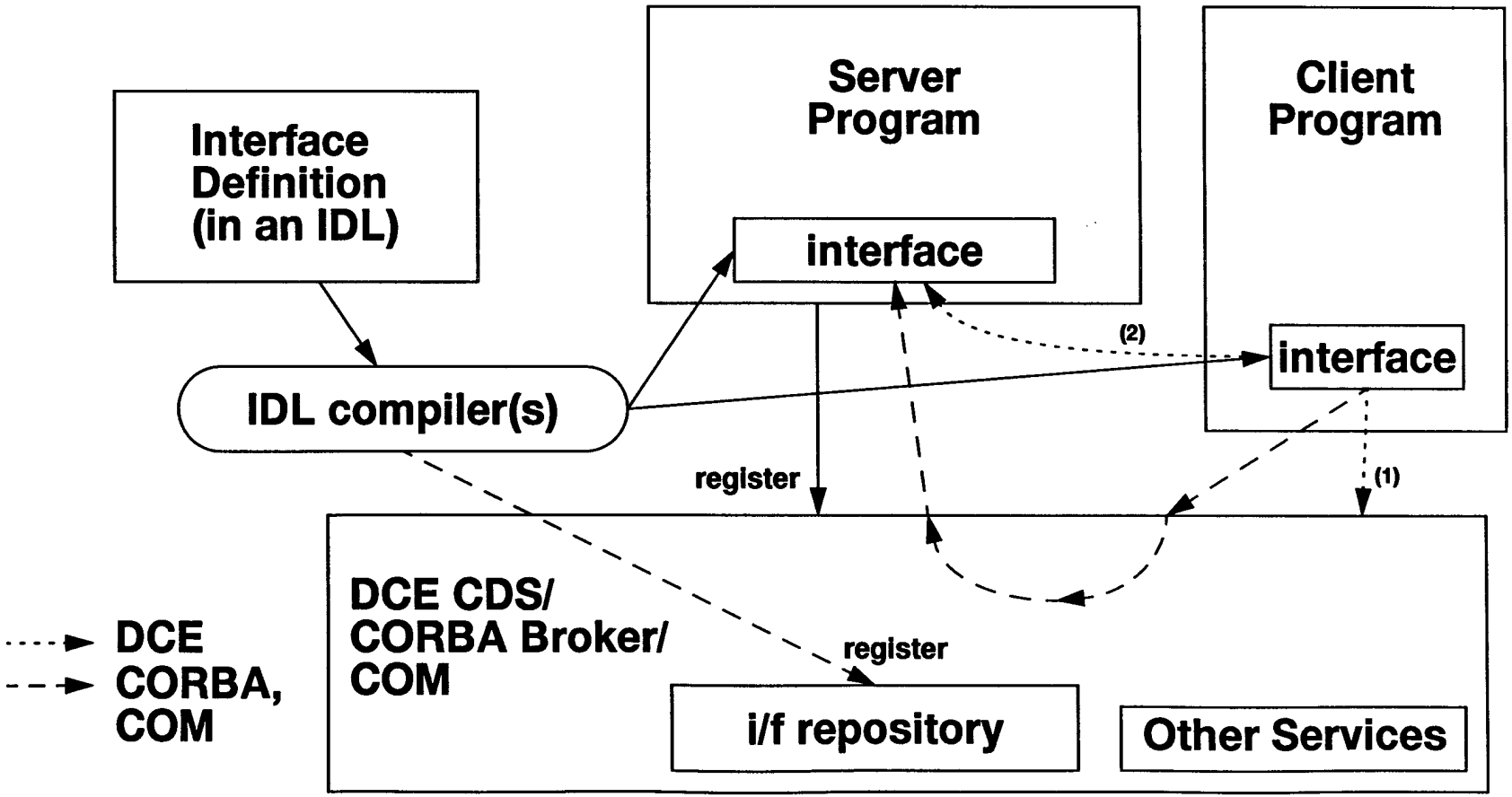
All of these different mechanisms have some points in common, so before discussing each it is useful to see their similarities.

In all systems programs may take on the role of a client, the role of a server, or both roles. A server provides some service, while a client requests a service from a server. To enable a client to make a request of a server, the server's interface is defined using an interface definition language (IDL). DCE, CORBA, and COM all have IDLs, although unfortunately they are not compatible. The interface (defined using the IDL) is then processed by an IDL compiler, which then automatically generates code that will become part of the server program and client program. This generated code handles the details such as bit ordering, data transmission protocols, and so on.

In both CORBA and COM, processing the interface definition also registers interface information in a repository so that other programs can later make queries about the interface.

At run-time, servers start up and register themselves with some centralized program (called the CDS in DCE, the broker in CORBA, and COM in COM). Clients may then make requests to the servers. In DCE, a client makes a request to CDS (the name server) to locate the server of interest and then directly calls the server. In CORBA and COM, the client makes a request to the broker or COM which then passes that request on to the server. Other services are also provided.

GENERAL APPROACH



IDA

DCE: BASIC CAPABILITIES

Now we will examine DCE's basic capabilities. DCE is based on the concept of a "remote procedure call" from one program to another. Programs request other programs to do something by "calling" a procedure and waiting for a reply, just as they do for an ordinary procedure in an ordinary procedural program. External interfaces of programs are defined in the DCE Interface Definition Language (IDL).

DCE also provides a number of additional services as shown. There are a large number of security-related services. Note that the "directory" service (needed for finding other services) may be replicated to improve reliability. Coordinated time is implemented by DCE to support security authentication time limits.

DCE users and systems are grouped into DCE "cells" for administration. Cells define domains of trust.

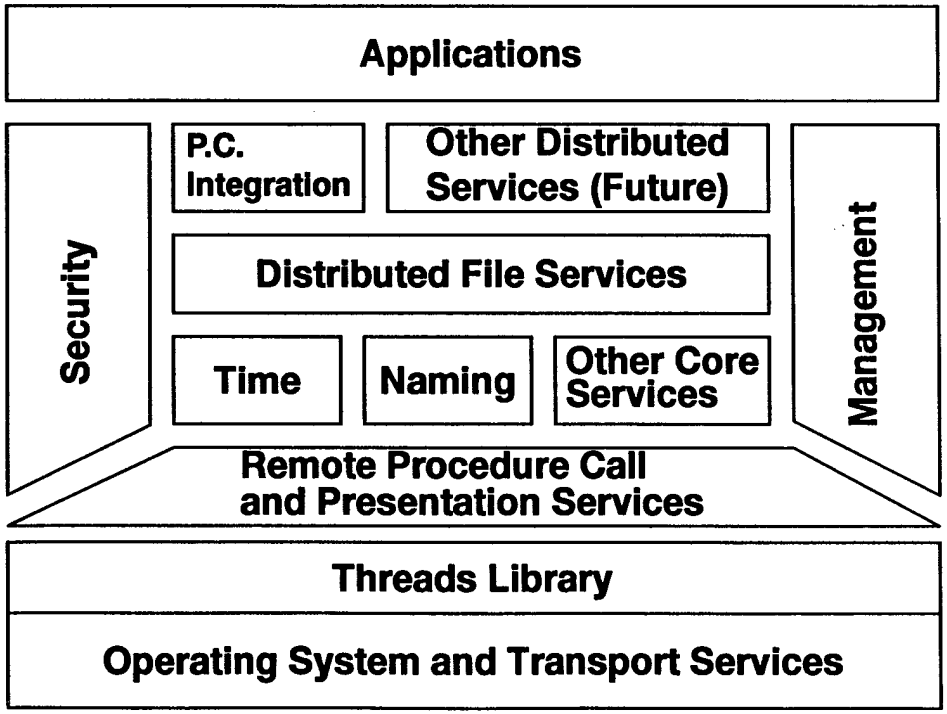
DCE: BASIC CAPABILITIES

- Based on remote procedure call (program-to-program)
  - Program interface defined in DCE IDL
  - C communications code generated from DCE IDL
- Provides additional useful standardized services:
  - Security (authentication, access control, encryption, time limits for authentication, delegation, auditing support)
  - Directory (find a service; DCE replicates for reliability)
  - Coordinated time (needed for security)
  - Thread interface (low-overhead tasks)
  - Distributed file system (sold separately)
- DCE users and systems grouped into DCE "cells" for administration
- Other services can be built on DCE (e.g. transactions, CORBA)

OSF DCE ARCHITECTURE

This is a standard picture used to show how DCE is implemented. Note in particular how nearly all other services are based on the remote procedure call services.

OSF DCE ARCHITECTURE



DCE: BACKER AND PRODUCTS

The Open Software Foundation (OSF) maintains a DCE reference implementation. Others port this implementation to various computers and add tools. Since all implementations build on a common base, interoperation of different DCE systems is relatively straightforward.

DCE: BACKER AND PRODUCTS

- Open Software Foundation (OSF) develops DCE reference implementation; X/Open develops DCE specification and runs test suite; Open Group owns both
- Others ("all but Sun") port DCE to various computers and add tools; common base makes interoperation easier. Key companies:
- TransArc Corporation:
  - wholly owned subsidiary of IBM
  - their DCE port is used by DII COE
  - "Encina" implements transactions on DCE
  - "DFS" implements a distributed (& replicated) file system on DCE (DFS was used by the 1996 Olympics Web servers)
- UnixPros: older DCE port, used by Army GCCS (mobile computing)

DCE: FUTURE PLANS

OSF is working on ways to support C++ and public key cryptography [Rosenberry 1993].

Note that CORBA can be implemented on top of DCE.

DCE: FUTURE PLANS

- To be added this year:
  - Support for C++ and OO based on HP's OODCE
  - Architecture supporting public key encryption (for stronger security); no commitment to implement
- Support for security service replication due January 1997
- Ada95 DCE IDL implementation being discussed with TransArc; Smalltalk and Visual Basic interfacing products exist
- CORBA can be implemented on top of DCE

CORBA: BASIC CAPABILITIES

CORBA is based on object-oriented concepts. In CORBA, objects request other objects to perform services for them. These requests go through an object request broker (ORB). The goal is to support a collection of distributed communicating objects. A large number of different objects may be implemented in a single program, which creates a distinction between CORBA and DCE. Objects may inherit interfaces from other objects, a key construct in object-oriented programming. Program interfaces are defined in the CORBA IDL (which is different than the DCE IDL).

"CORBA" is actually the specification for an ORB, and is part of a larger architecture termed the Object Management Architecture (OMA). The OMA contains the following components: an ORB (as specified in CORBA), object services (fundamental capabilities for realizing and maintaining objects), common facilities (less fundamental capabilities useful in many applications), and domain interfaces (functions useful in specific domains) [OMG 1996]. However, the term "OMA" is not as widely known as the term "CORBA", and many use the term "CORBA" as a synonym for "OMA." In this presentation, the term OMA will occasionally be written as "OMA (CORBA)." The OMA object services are also called CORBAservices, and the common facilities are also called CORBAfacilities.

iCORBA supports a number of different languages, including C, C++, Ada95, Smalltalk, and Java. Version 2.0 of CORBA added a specification for ORBs to communicate with each other; before this only programs communicating with the same ORB could communicate with each other. There is a simple required inter-ORB interface that all ORBs must support, plus an optional interface for ORBs that are both DCE-based (to support DCE's capabilities such as security).

The CORBA specification provides a number of useful communication features, including asynchrony (sending a message without waiting for a reply) and dynamic invocation (being able to determine at run-time the interface of a component, and then making a request to that component). Note that at this time asynchrony is only supported by the dynamic interface (described later).

A rich set of object services and common facilities are planned, but specifications are just being completed for some services and facilities. OMA (CORBA) implementations tend to implement few services beyond the basic ORB at this time. The security specifications are flexible but incomplete; additional security specifications are being developed.

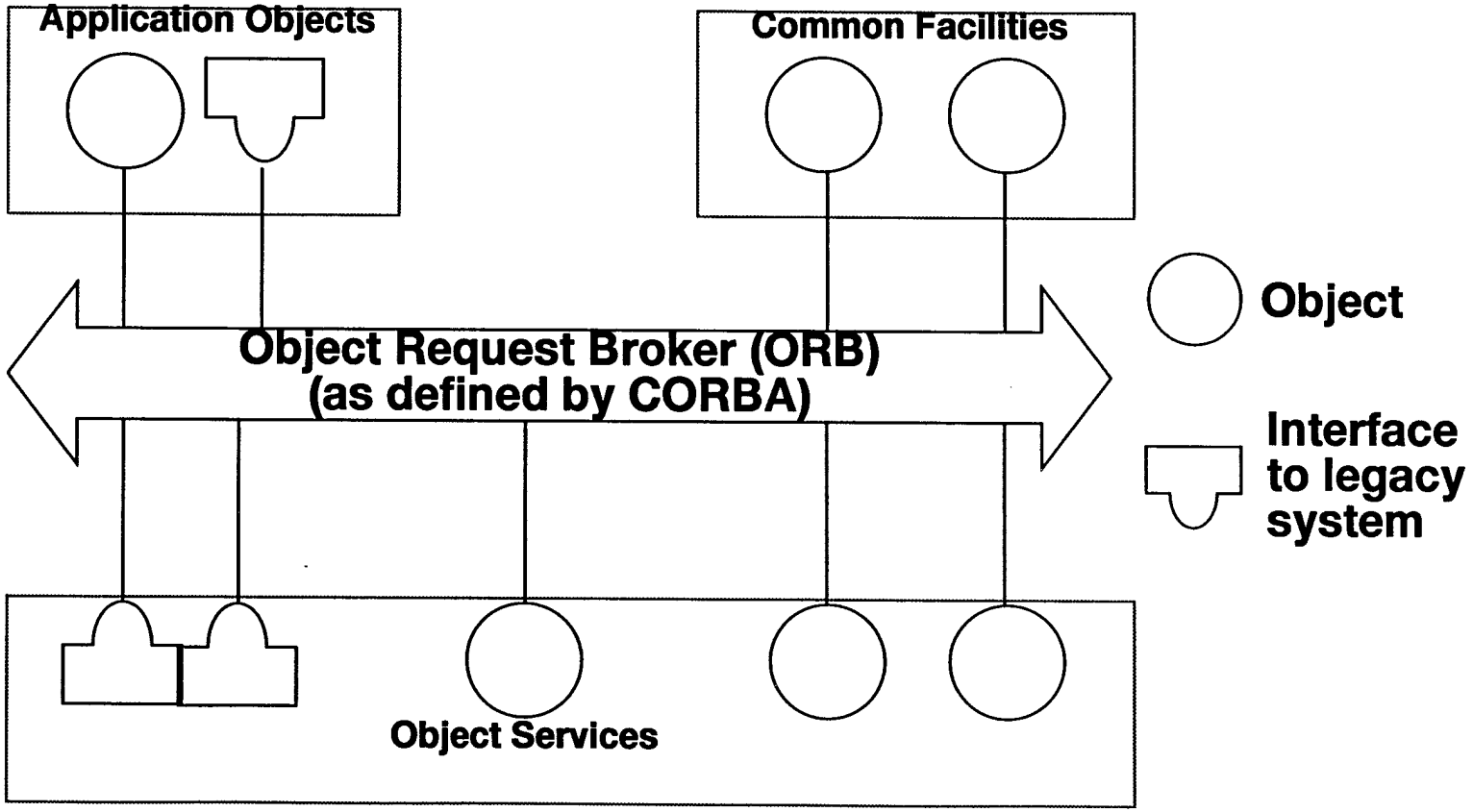
CORBA: BASIC CAPABILITIES

- Based on object-orientation (object-to-object, interface inheritance)
  - Objects request other objects to perform tasks via an ORB; the ORB locates the remote object and makes the request
  - Goal: Distributed object system
  - Program interface defined in CORBA IDL (generates code)
- OMA = ORB+object services+common facilities+domain interfaces
- Strengths and Weaknesses
  - Supports many languages (C, C++, Smalltalk, Ada95; Java)
  - CORBA 2.0 added specifications for ORB interoperability: a simple required one plus optional DCE-based
  - Supports asynchrony and dynamic invocation
  - Rich set of supporting services and facilities planned, but few currently specified and even fewer implemented
  - Security weak: first specification just released; more to come

OBJECT MANAGEMENT ARCHITECTURE (OMA)

This is a standard diagram of the Object Management Architecture (OMA). Applications can request object services and common facilities using the object request broker (ORB). The ORB must conform to the CORBA specification. Some of these "objects" may be legacy systems with a "wrapper" defining its interface. Not shown are domain interfaces, which would look the same architecturally as the common facilities.

OBJECT MANAGEMENT ARCHITECTURE (OMA)



Object Services - Basic functions for realizing and maintaining objects
Common Facilities - Collection of classes and objects useful in many applications
Application Objects - Specific to particular end user applications

OMA (CORBA): BASIC CAPABILITIES: PLANNED SERVICES AND FACILITIES

This chart shows the planned OMA object services (also called CORBAservices) and OMA common facilities (also called CORBAfacilities). All of these services and facilities must be requested via the ORB (as defined by CORBA).

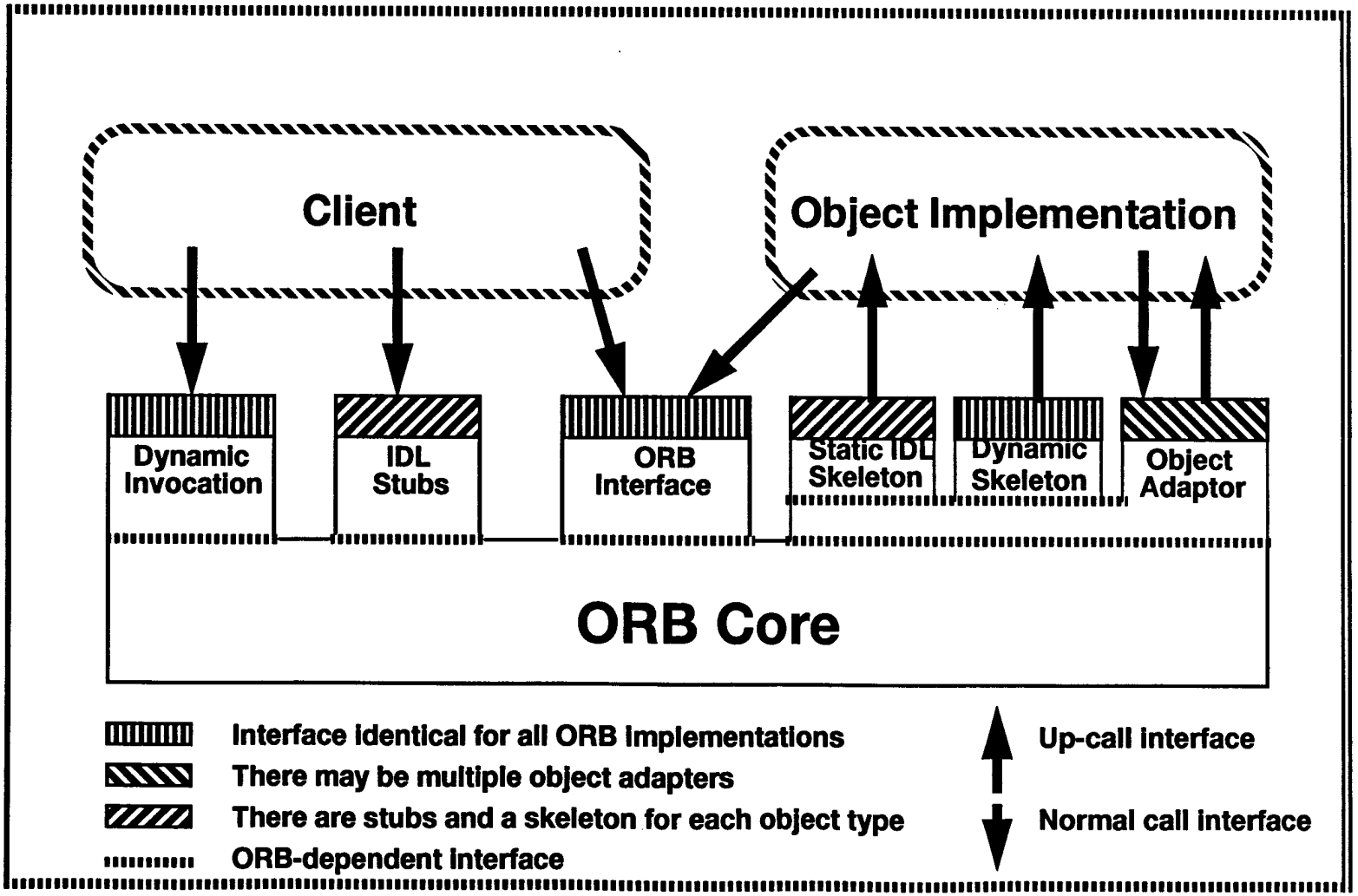
OMA (CORBA): BASIC CAPABILITIES: PLANNED SERVICES AND FACILITIES

- CORBAservices™ (considered "fundamental"):
  - Naming (binds names to objects)
  - Event Management (receive notification e.g. of changes)
  - Persistent Object (managing persistent data)
  - Lifecycle (creating, deleting, copying, moving objects)
  - Concurrency Control
  - Externalization (import/export objects)
  - Relationship (store relationships between objects)
  - Transactions (commit or abort a series of actions)
  - Query (relationships/properties of objects)
  - Security: adopted specification generic; second one in process
- CORBAfacilities™ ("not as fundamental")—work ongoing
  - UI, Information Management (includes time), system management, task management; OpenDoc accepted

CORBA 2.0: DETAILS ON CLIENT, SERVER, ORB

This is a standard diagram showing how clients and servers (the latter is labelled "object implementation") interact in CORBA. If clients know the server's interface at client compile time, the client can use the "IDL stubs" generated by the IDL processor. If a client does not know the server's interface, it can still make requests to a server using dynamic invocation. Note that both the client and server can make requests to the ORB for various services.

CORBA 2.0: DETAILS ON CLIENT, SERVER, ORB



IDA

CORBA IDL EXAMPLE

This is a trivial example of CORBA's interface definition language (IDL); in some sense it is also an example of how all of these interface definition languages work.

This example shows a trivial interface to a bank account. Each account has a balance, whose value can be read (but not arbitrarily changed, so that attribute is marked as "readonly"). There are two operations permitted on an account: "makeDeposit" and "makeWithdrawal." These operations have additional parameters as shown.

CORBA IDL EXAMPLE

```
// Trivial CORBA IDL example
interface Account {
readonly attribute float balance;

void makeDeposit(in float sum);
void makeWithdrawal(in float sum, out float newBalance);
};
```

OMA AND CORBA: BACKER AND PRODUCTS

OMA and CORBA are defined and backed by the Object Management Group (OMG). The general approach of the OMG has been to send out request for proposals (RFPs) to industry; specifications are then developed through consensus.

Many different implementations with different starting points are a result. This has the advantage of providing more differentiated products, and the disadvantage of resulting in products that may have less interoperability and commonality.

More information on CORBA can be retrieved from OMG at "http://www.omg.org/". CORBA is defined and further discussed in various documents from the OMG [1995a, 1995b, 1996a, 1996b].

OMA AND CORBA: BACKER AND PRODUCTS

- Backer: Object Management Group (OMG)
- Specifications developed through vendor consensus. Vendors then build their own.
- Many different implementations with different starting points, e.g.:
  - IONA Orbix. Supports C++, Ada95, and Java on many platforms. Supports IIOP. Special capabilities: filters, interface with OLE. To come: event, transactions, IBM SOM interaction.
  - IBM SOM/DSOM. Included with OS/2, also for Windows and AIX. Supports Ada95. Special capabilities: meta-object capability, binary compatibility. DSOM uses DCE. Fall: faster.
  - HP ORB Plus. DCE-based on HP-UX.
  - Digital ObjectBroker. Working to integrate CORBA and Microsoft's OLE/COM ("COM*"), using modified DCE.
  - Sun NEO (formerly DOE). C++, Java. Special: persistence.
  - Many others (Expersoft PowerBroker, VisiGenic VisiBroker)

OMA (CORBA): FUTURE PLANS

A great deal of work is currently underway. One example is the area of security; work is ongoing to define a security service and a protocol for secure interoperation between ORBs.

Another area of work is to somehow integrate with Microsoft's ActiveX, OLE, and COM. COM is the subject of the next charts.

OMA (CORBA): FUTURE PLANS

- Security and security interoperability
  - current security specification is a generic framework
  - work ongoing to define a security service and secure ORB interoperability; initial comments promising
- Revision of IIOP (ORB interoperability) and ESIOP (DCE ORB interoperability) underway to improve interoperability
- Complete work on specifying (and implementing) services and facilities
- Work to integrate with Microsoft's ActiveX, OLE, and COM

MICROSOFT COM

The Component Object Model (COM) is Microsoft's approach to application intercommunication, and is the basis of OLE (described later). COM permits programs to make requests to "COM objects" via predefined interfaces. A COM object usually implements a large number of interfaces, and must implement a special interface that permits other programs to query it and learn which other interfaces it supports.

Unlike classical object-oriented systems (such as what CORBA supports), the COM has no concept of a "unique identifier" for a specific object. In addition, COM does not support inheritance. Thus, COM is considered an "object-based" approach rather than an "object-oriented" approach.

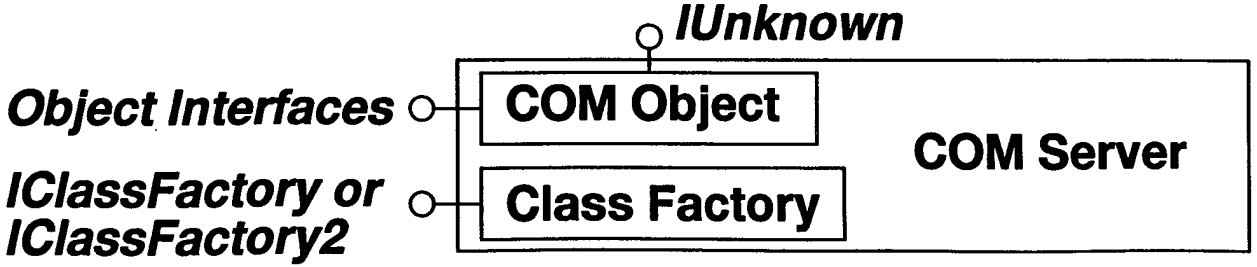
In COM, a user-defined interface is termed a "custom" interface. COM objects implement one or more interfaces. Clients can query a COM object using the "IUnknown" interface to determine if a given COM object supports a given interface. Interfaces cannot be extended later; to add a new operation you must define another interface. For example, the "IClassFactory" interface is used to create new instances of an object, but there are actually two such interfaces; IClassFactory2 was added later to support software licensing.

COM servers are pieces of code that house one or more object classes. When a client asks for an object with a specific class (a "CLSID"), COM loads the server code and asks the code to create an object of that class. COM servers must implement the IClassFactory interface.

COM objects are also called OLE objects or Windows objects [Orfali 1996].

MICROSOFT COM

- Microsoft Component Object Model (COM) = Windows' inter-application communication mechanism (broker)
- Based on an object-based (not OO) approach
  - An interface defines a contract between COM objects
  - New ("custom") interfaces can be user-defined
  - COM servers implement one or more interfaces
  - Clients can determine which interfaces are supported by a server and call those interfaces (static or dynamic invocation)
  - Interfaces cannot be extended later (no inheritance or versioning); to add new operations, define another interface
  - Interfaces are unique, but objects don't have an ID



IDA

MICROSOFT COM: RELATED PRODUCTS AND FUTURE PLANS

Related terms include OLE 1, OLE 2 (now called OLE), and ActiveX; since these terms can be confusing, definitions are provided.

Distributed COM (DCOM) provides COM between platforms and is shipping with Windows NT 4.0; it is to eventually be available on Windows 95. Work is ongoing to make COM available on Macintoshes and Unix.

Microsoft and DEC have proposed an approach for interoperating Microsoft's COM and CORBA. This approach is termed the Common Object Model, which is also called "COM". To distinguish the joint Microsoft/DEC COM from Microsoft's COM, this document will call the joint work "COM*" (a convention used by others). COM* permits OLE objects to interact with CORBA server objects on platforms running DEC's ObjectBroker. Note that this is primarily limited to a unidirectional call. OMG is also working with Microsoft on more general approaches to connecting CORBA and Microsoft's COM.

Microsoft has announced that it will release ActiveX technologies from their proprietary status to the Open Group. This would include COM and DCOM. The ramifications of this change are unclear at this time.

MICROSOFT COM: RELATED PRODUCTS AND FUTURE PLANS

- Related terms/products:
  - OLE 1 (1990): "Object Linking and Embedding"; Clumsy compound document approach.
  - OLE 2 (1993), now called OLE: New approach for inter-application coordination based on COM. Intent: Windows will evolve into a giant set of OLE services.
  - ActiveX (1995): slimmed-down set of OLE and OLE-like services for Web applications; also based on COM
- Distributed COM (DCOM) provides COM between platforms, shipping with Windows NT 4.0 and eventually Windows 95.
- Work ongoing to make COM available on Macintoshes and Unix and to interoperate with CORBA.
- Microsoft has announced its intention to release ActiveX technologies from its proprietary status to the Open Group. This includes COM and DCOM.

CONTRAST: MICROSOFT COM VS. CORBA

There are number of weaknesses in Microsoft COM when compared to CORBA. Microsoft is currently working to expand COM support to non-Microsoft platforms, but COM is still primarily for systems running Microsoft operating systems; by contrast, CORBA implementations are available on a wide variety of operating systems. COM is technically weaker in several areas; it is not as object-oriented (which can be a weakness when dealing with object-oriented systems) and does not support security nor versioning.

One especially odd weakness of COM is that interfaces must be defined twice for each system and kept consistent manually. This is because interfaces are defined using two different interface languages loosely based on DCE's IDL, termed the IDL and the object description language (ODL).

The COM is only specified as a binary interface, which is easier to use from Microsoft C++ but can be difficult to use from other compilers and languages depending on how they implement their internal data structures and their capabilities for specifying low-level interfaces [Orfali 1996, 434].

As noted earlier, COM and OLE are currently proprietary, though there has been a stated intent to move them to the Open Group.

Microsoft's formidable market strengths make COM very important. Microsoft dominates the desktop platform, COM and OLE are already deployed in Windows systems, and most major applications already support COM and OLE, including tool category market leaders. It is possible that CORBA and DCE implementations could be overtaken by COM, resulting in more difficult-to-maintain CORBA or DCE infrastructures.

CONTRAST: MICROSOFT COM VS. CORBA

- Weaknesses of Microsoft COM (Microsoft's ORB approach):
  - Primarily single platform
  - Technically weaker (not OO, no security, no versioning, 2x IDL)
  - Binary standard makes it less language-independent
  - Currently proprietary and vendor-locked (but stated intention to change)
- Strengths:
  - Microsoft's dominance of desktop platforms
  - COM already in Windows and is high-speed without networks
  - OLE already supported by many applications, including market dominators. OLE (and ActiveX) is based on COM
- Issues:
  - Could crush market, resulting in expensive unmaintained CORBA ORBs and DCEs

CONTRAST: DCE VS. CORBA

DCE's strengths include its security features, interoperability, and maturity. CORBA's strengths include its object-orientation and support for many languages.

Note selecting an interfacing mechanism is not enough; interfaces using these mechanisms must be defined for the mechanism to be useful.

CONTRAST: DCE VS. CORBA

- DCE Strengths:
  - Security features
  - Interoperable implementations (based on same source)
  - Mature: All implementations support same basic feature set, replication increases reliability
  - Enterprise orientation
- CORBA Strengths:
  - Object-oriented: easier to use, esp. with OO programs
  - Stronger support for non-C/C++ (Ada, Smalltalk, Java)
  - Specification supports dynamic invocation and asynchrony
- Selecting DCE, CORBA, or COM is not enough; interfaces using them must be defined
- Either can support developing highly coordinated BMC/3 systems

CORBA IMMATURITY

A key issue in using CORBA is the maturity of CORBA implementations. The sufficiency of CORBA implementations depends, naturally enough, on the expected use. CORBA compliant ORBs exist now, with performance sufficient for lower volume (operations/second) use; higher volume use is questionable. Here are documented user testimonials and performance evaluations that we were able to obtain.

MITRE's DOMIS CTAPS work found that, when very coarse-grained objects used CORBA, the performance was quite acceptable. Note that their requirements were quite modest, simply a reply in less than one second [Brando 1994, 1995a, 1995b]. Gokhale and Schmidt have performed performance experiments with CORBA implementations on high-speed networks; their work is described in [Gokhale 1996].

ISSUE: CORBA IMMATURITY

- CORBA "immaturity" depends on the expected use. CORBA ORBs exist now, with performance sufficient for lower volume (operations/second) use; higher volume use questionable
  - DOMIS CTAPS (MITRE 1994): performance OK (coarse-grain, <1s response)
  - DARPA JTF ATD: OK for research application (planning and situation assessment). Orbix much slower than C++ method call on single platform. "Everyone's waiting to see how good IBM SOM's implementation will be (in Fall 1996)"
  - DMSO HLA: CORBA too slow for very large number of (simulation) objects. Will primarily use custom C++ interface; CORBA interface will be maintained
  - Gokhale and Schmidt (WA University @ St. Louis): currently, significant overhead specific to high-speed (Gigabit+) networks (75-80% scalar, 30% struct vs. custom calls)

ISSUE: CORBA IMMATURITY (CONTINUED)

The AT&T experience is further described in [Carlson 1996].
SEI's performance results are documented in [Wallnau 1995].

ISSUE: CORBA IMMATURITY (CONTINUED)

- DII COE: Claimed intent is to move to DCE, then define APIs using CORBA yet implement APIs on top of DCE. Unclear if this will really occur, many are considering the necessity and cost of DCE and CORBA. DCE included in DII COE 2.0, not yet used
- AT&T ISTEL: CORBA OK for life insurance quotation system involving 50 Windows NT servers integrating 50 different companies with <10s response time. Demand unspecified
- SEI: CORBA implementations 1/4 speed of specially-built communication (using sockets) between machines, but (surprisingly) faster than Sun RPC

At this time it is difficult to argue that any one of these mechanisms should be mandated for all purposes. All impose communication overhead which may be unacceptable in some contexts. At this time the commercial "winner" is uncertain.

Within a BM/C3 system and between strategic (NMD) BM/C3 systems, these mechanisms may well be acceptable if the request rate (requests/second) are kept low. Between TMD systems they are quite questionable since large bandwidth cannot be guaranteed. Naturally, simulation studies are necessary to evaluate the suitability of any particular use.

There are other possible uses of CORBA and DCE (and possibly COM). For example, the interface to a communication module (say for link-16) could be defined using a DCE or CORBA interface. Another possibility is using IDL (probably CORBA IDL) as a design "notation" for defining module interfaces.

BMDO ISSUES

- Should any mechanism be mandated, and for what?
  - All impose communication overhead
  - "Winner(s)" of DCE vs. CORBA vs. COM uncertain
- From a performance viewpoint:
  - Within a BM/C3 system: very likely to be acceptable, especially if request rate (requests/second) kept low
  - Between strategic (NMD) BM/C3s: likely to be acceptable
  - Between TMD systems: questionable since large bandwidth cannot be guaranteed.
- Other uses of CORBA/DCE:
  - Communication module with DCE/CORBA interface
  - Use IDL as design "notation"
- Recommendation: study requirements and prototype

IDA

APPENDIX A SUMMARY

In summary: DCE is based on remote procedure calls and includes features such as security. CORBA is based on object-orientation and provides stronger support for different computer languages, but is less mature than DCE. COM is Microsoft's approach, with some heavy commercial support but wide cross-platform support is not yet available.

Selecting an interfacing mechanism is not enough; interfaces using these mechanisms must be defined for the mechanism to be useful. With careful use, these mechanisms might aid BMDO in certain areas.

APPENDIX A SUMMARY

- DCE
  - Based on remote procedure call (program-to-program)
  - Includes security features, time synchronization, directory
  - Other services can be built on DCE, e.g. transactions, CORBA
- CORBA
  - Based on object-orientation (object-to-object)
  - Stronger support for non-C/C++ (Ada, Smalltalk, Java)
  - Specification supports dynamic invocation and asynchrony
  - Less mature than DCE
- COM
  - Microsoft support; wide cross-platform to be available
- Selecting one is not enough; interfaces using them must be defined
- Could support highly coordinated BM/C3 with careful use

ACRONYMS

ABIS..... Advanced Battlespace Information System
ADTOC ..... Air Defense Tactical Operations Center
ALERT..... Attack and Launch Early Report to Theater
AOC..... Air Operations Center
API..... Application Programmer Interface
AT&T ..... American Telephone and Telegraph
ATD ..... Advanced Technology Demonstration
ATDL ..... Army Tactical Data Link
ATM..... Asynchronous Transfer Mode
AWACS ..... Airborne Warning and Control System
BMC3..... Battle Management, Command, Control, and Communications
BMD..... Ballistic Missile Defense
BMDO ..... Ballistic Missile Defense Organization
Batt..... Battalion
Bgde..... Brigade
CA ..... Circa
C2 ..... Command and Control
C4I..... Command, Control, Communications, Computers, and Intelligence
C4IFW ..... C4I for the Warrior
C4ISR ..... Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CDE..... Common Desktop Environment
CDS..... Cell Directory Service
CINC ..... Commander in Chief
CM..... Configuration Management
CNR..... Combat Net Radio
COE..... Common Operating Environment
COEA ..... Cost and Operational Effectiveness Analysis (or Assessment)
COM ..... Component Object Model
COM*..... Common Object Model
CORBA ..... Common Object Request Broker Architecture
COTS ..... Commercial Off-The-Shelf
CPU..... Central Processing Unit
CRC..... Control and Reporting Center
CTAPS ..... Contingency Theater Automated Planning System
CV/LHD/LCC... Carrier/amphibious helicopter carrier/ command ship
DARPA ..... Defense Advanced Research Projects Agency
DBS..... Direct Broadcast Service
DCE..... Distributed Computing Environment
DCOM ..... Distributed Component Object Model (COM)
DDDS ..... Defense Data Dictionary System

ACRONYMS (CONTINUED)

DEC . . . . . Digital Equipment Corporation
DFS . . . . . Distributed File System
DII . . . . . Defense Information Infrastructure
DISA. . . . . Defense Information Systems Agency
DMSO . . . . . Defense Modeling and Simulation Office
DOE . . . . . Distributed Objects Everywhere (see NEO)
DOMIS . . . . . Distributed Object Management Integration System
DSOM . . . . . (IBM) Distributed SOM
DoD . . . . . Department of Defense
EW. . . . . Early Warning
FDDI. . . . . Fiber Distributed Data Interface; Fiber-optic Digital Data Interchange
FORTRAN . . . . . Formula Translation
FPTOC. . . . . Field Projection Tactical Operations Center
FY . . . . . Fiscal Year
GBDL. . . . . Ground-Based Data Link
GBI. . . . . Ground Based Interceptor
GBR. . . . . Ground Based Radar
GBS . . . . . Global Broadcast Service
GCCS. . . . . Global Command and Control
GSS . . . . . Generic Security Services
GUI. . . . . Graphical User Interface
HAWK . . . . . Homing All-the-Way Killer
HCI . . . . . Human Computer Interaction
HLA . . . . . High-Level Architecture (for simulation)
I&RTS . . . . . Integration and Run-Time Specification
IBM . . . . . International Business Machines
IDA . . . . . Institute for Defense Analyses
IDEF . . . . . Information Definition #1; ICAM Definition Language
IDL. . . . . Interface Definition Language
IIOP . . . . . Internet Inter-ORB Protocol
IP. . . . . Internet Protocol
ISTEL . . . . . (AT&T) Obsolete name of Lanner Group Business Center
JCTN . . . . . Joint Composite Tracking Network
JDN . . . . . Joint Data Network
JMTK. . . . . Joint Mapping Toolkit
JPN . . . . . Joint Planning Network
JTA . . . . . Joint Technical Architecture
JTF . . . . . Joint Task Force
JTIDS . . . . . Joint Tactical Information Distribution System
LAN . . . . . Local Area Network
MCG&I. . . . . Mapping, Charting, Geodesy & Intelligence
MIDS . . . . . Multifunctional Information Distribution System

ACRONYMS (CONTINUED)

MIL-STD. . . . . Military Standard
MILSATCOM . . . Military Satellite Communication
NAD . . . . . Navy Area Defense
NEO . . . . . Sun product, formerly DOE; not an acronym.
NMD. . . . . National Missile Defense
NSA . . . . . National Security Agency
OA . . . . . Operational Architecture (see TA, SA)
ODBC. . . . . Open Database Connectivity
ODL . . . . . Object Description Language (see IDL)
OLE . . . . . (originally) Object Linking and Embedding
OMA. . . . . Object Management Architecture
OMG . . . . . Object Management Group
OO . . . . . Object-Oriented
OODCE . . . . . Object-Oriented DCE
ORB. . . . . Object Request Broker
OSD . . . . . Office of the Secretary of Defense
OSF . . . . . Open Software Foundation
PAC-3 . . . . . Patriot Advanced Capability-3
PADIL. . . . . PATRIOT Digital Information Link
POSIX . . . . . Portable Operating System Interface for Computer Environments
PROGRUS. . . . . PROGram Update Study
RDBMS . . . . . Relational Database Management System
RFP . . . . . Request for Proposal
ROE. . . . . Rules of Engagement
SA . . . . . System Architecture (see OA, TA)
SBIRS . . . . . Space-Based Infrared System
SBWS . . . . . Space-Based Warning System
SBWS . . . . . Space-Based Warning System
SGI . . . . . Silicon Graphics Incorporated
SOM . . . . . (IBM) System Object Model
SONET . . . . . Synchronous Optical Network
SQL. . . . . Structured Query Language
TA . . . . . Technical Architecture (see OA, SA)
TACC. . . . . Tactical Air Command Center
TAD . . . . . Theater Area Defense
TADIL . . . . . Tactical Digital Information Link
TAFIM . . . . . Technical Architecture Framework for Information Management
TAOC . . . . . Tactical Air Operations Center
TCP/IP. . . . . Transmission Control Protocol/Internet Protocol
TDDS. . . . . TRAP Data Dissemination System
TFCC. . . . . Tactical Flag Command Center
TIBS . . . . . Tactical Information Broadcast Service
TMD. . . . . Theater Missile Defense
TRAP. . . . . Tactical Related Applications
TRI-TAC . . . . . Tri-Services - Tactical Communications
TRM. . . . . Technical Reference Model

ACRONYMS (CONTINUED)

UHF . . . . . Ultrahigh Frequency
UI . . . . . User Interface
US . . . . . United States
VMF . . . . . Variable Message Format
WA . . . . . Washington
WABI . . . . . Windows Application Binary Interface
WG . . . . . Working Group
WWMCCS . . . . . World-Wide Military Command and Control System
Win32 . . . . . (Microsoft) Windows, 32-bit interface

REFERENCES

[Brando 1994] Brando, Thomas J. December 1994. *DOMIS Implementation of CTAPS Functionality Using Orbix*. MITRE Document MP 94B-287. MITRE. "http://www.mitre.org/research/domis/reports/Orbix.html".

[Brando 1995a] Brando, Thomas J. March 1995. *Comparing DCE and CORBA*. MITRE Document MP 95B-93. MITRE. "http://www.mitre.org/research/domis/reports/DCEvCORBA.html".

[Brando 1995b] Brando, Thomas J. September 1995. *Distributed Object Management Integration System (DOMISC) Final Report*. MITRE Document MP95B-320. MITRE. "http://www.mitre.org/research/domis/reports/finrep94.ps".

[BMDO 1996] Ballistic Missile Defense Organization (BMDO). August 1996. *Theater Missile Defense Command and Control Plan* (C2 Plan). Previously published on 31 October 1995 as a coordination draft.

[Carlson 1996] Carlson, Andrew C., William R. Brook, Christopher L.F. Haynes. 1996. *Experiences with Distributed Objects*. AT&T Technical Journal. March/April 1996. pp. 58-67.

[Comer 1991] Comer, Douglas E. 1991. *Internetworking with TCP/IP Vol I: Principles, Protocols, and Architecture*. Second Edition. ISBN 0-13-468505-9. Englewood Cliffs, NJ: Prentice Hall.

[CCP 1996] Common Criteria Project. 1996. Common Criteria for Information Technology Security Evaluation. Available at "http://www.itsec.gov.uk".

[DDR&E 1995] DDR&E. 19 July 1995. *DDR&E Science and Technology Review.*

[DoD 1996] Department of Defense. August 22, 1996. *Joint Technical Architecture*. Version 1.0. The document and related content are available at URL "*http://www.itsi.disa.mil/jta.html*".

[Gokhale 1996] Gokhale, Aniruddha, and Douglas C. Schmidt. August 1996. *Measuring the Performance of Communication Middleware on High-Speed Networks*. SIGCOMM Conference, ACM 1996, Stanford University, August 28-30, 1996. "http://www.cs.wustl.edu/~schmidt/corba-research.html"

[Golliday 1985] Golliday, C. Leslie, September 1985, "Data Link Communications in Tactical Air Command and Control Systems." *IEEE Journal on Selected Areas in Communications*. Vol SAC-3, No. 5.

[Goralski 1995] Goralski, Walter J. 1995. *Introduction to ATM Networking*. New York: McGraw-Hill.

[I&RTS 1995] Joint Interoperability and Engineering Organization (JIEO) Defense Information Systems Agency (DISA). October 23, 1995. *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS).* DII COE documents are available at "*http://spider.osfl.disa.mil/dii/*".

[Jacobson 1990] Jacobson, V. February 1990. *Compressing TCP/IP Headers for Low-Speed Serial Links.* RFC 1144. Available at "ftp://nic.ddn.mil/rfc/rfc1144.txt".

[Logicon 1994] Logicon, Inc. April 1994. *Understanding Link-16: A Guidebook for New Users.* San Diego, CA: Logicon, Inc. Tactical and Training Systems.

[MNS 1995] May 1995. *Mission Needs Statement for Global Command and Control System (GCCS).* GCCS documents are available at URL "http://164.117.208.50/".

[NRC 1991] National Research Council. 1991. *Computers at Risk: Safe Computing in the Information Age.* Washington, DC: National Academy Press.

[OMG 1995a] Object Management Group (OMG). July 1995. *The Common Object Request Broker: Architecture and Specification.* Revision 2.0. Framingham, MA: Object Management Group.

[OMG 1995b] Object Management Group (OMG). November 1995. *Common Facilities Architecture.* Revision 4.0. Framingham, MA: Object Management Group.

[OMG 1996a] Object Management Group (OMG). Updated March 28, 1996. *CORBAservices: Common Object Services Specification.* Framingham, MA: Object Management Group.

[OMG 1996b] Object Management Group. 1996. *OMA Executive Overview.* URL "http://www.omg.org/omaov.htm".

[Orfali 1996] Orfali, Robert, Dan Harkey, and Jeri Edwards. 1996. *The Essential Distributed Objects Survival Guide.* New York: John Wiley and Sons. ISBN 0-471-12993-3.

[Postel 1980] Postel, Jon. 28 August 1981. *User Datagram Protocol.* RFC 768. Available at "ftp://nic.ddn.mil/rfc/rfc768.txt".

[Postel 1981a] Postel, Jon. September 1981. *Internet Protocol: DARPA Internet Program Protocol Specification.* RFC 791. Available at "ftp://nic.ddn.mil/rfc/rfc791.txt".

[Postel 1981b] Postel, Jon. September 1981. *Transmission Control Protocol.* RFC 793. Available at "ftp://nic.ddn.mil/rfc/rfc793.txt".

[Wallnau 1995] Wallnau, Kurt C., and John Rice. 1995. "ORBs in the Midst: Studying a New Species of CASE Integration Mechanism." *Proceedings of the Seventh International Workshop on Computer-Aided Software Engineering, July 10-14, 1995, Toronto, Ontario, Canada.* Los Alamitos, CA: IEEE.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Final

4. TITLE AND SUBTITLE
FY96 Analysis of the Ballistic Missile Defense Interoperability Standards

5. FUNDING NUMBERS
DASW01-94-C-0054

Task Order T-R2-597.2

6. AUTHOR(S)
Dennis W. Fife, Reginald N. Meeson, David A. Wheeler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses (IDA)
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-3277

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Ballistic Missile Defense Organization (BMDO)
The Pentagon, Room 1D110
Washington, DC 20301-7100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; unlimited distribution: 28 April 1997.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)

This briefing presents an assessment of the need for information technology interoperability standards and recommends actions that the Ballistic Missile Defense Organization (BMDO) can take based on that assessment. It highlights Issues in using the DoD Joint Technical Architecture (JTA) for the Ballistic Missile Defense (BMD) Battle Management, Command, Control and Communications (BMC3). These issues were identified by examining selected joint BMD scenarios, the standards needed to support these scenarios, and the state of relevant standards and suites of standards. The recommendations are divided into three areas: (1) standards selections and tracking, (2) interoperability architecture development, and (3) compliance and migration planning. Significant effort should be expended to monitor the maturity and usability of the relevant JTA standards and emerging technologies, and certain key standards and products should be accelerated. The BMD technical architecture should be extended to identify where and how such standards will be applied. Computer-to-computer transactions and application-level protocols should be defined and standardized for open system communications between BMD systems, separating the communications messages from the medium that carries them. BMDO and the Services should coordinate their compliance efforts, and note that compliance is an evolutionary process that should be based on long-term criteria.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Ballistic Missile Defense (BMD), Information Technology Standards, Open Systems, Scenarios, Joint Technical Architecture (JTA), Theater Missile Defense (TMD), Battle Management, Command, Control and Communications (BMC3).	152
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL